talend | Connecting the Data-Driven Enterprise

Composants Talend Open Studio

Guide de référence

5.6.3

Convient à la version 5.6.3. Annule et remplace toute version antérieure de ce guide.

Date de publication : 2 février 2016

Copyleft

Cette documentation est mise à disposition selon les termes du Contrat Public Creative Commons (CPCC).

Pour plus d'informations concernant votre utilisation de cette documentation en accord avec le Contrat CPCC, consultez : http://creativecommons.org/licenses/by-nc-sa/2.0/

Mentions légales

Tous les noms de marques, de produits, les noms de sociétés, les marques de commerce et de service sont la propriété de leurs détenteurs respectifs.

Licence applicable

Le logiciel décrit dans cette documentation est soumis à la Licence Apache, Version 2.0 (la "Licence"). Vous ne pouvez utiliser ce logiciel que conformément aux dispositions de la Licence. Vous pouvez obtenir une copie de la Licence sur http://www.apache.org/licenses/LICENSE-2.0.html (en anglais). Sauf lorsqu'explicitement prévu par la loi en vigueur ou accepté par écrit, le logiciel distribué sous la Licence est distribué "TEL QUEL", SANS GARANTIE OU CONDITION D'AUCUNE SORTE, expresse ou implicite. Consultez la Licence pour connaître la terminologie spécifique régissant les autorisations et les limites prévues par la Licence.

Ce produit comprend les logiciels développés par AOP Alliance (standards Java/J2EE AOP), ASM, Amazon, AntlR, Apache ActiveMQ, Apache Ant, Apache Axiom, Apache Axis, Apache Axis 2, Apache Batik, Apache CXF, Apache Chemistry, Apache Common Http Client, Apache Common Http Core, Apache Commons, Apache Commons Bcel, Apache Commons JxPath, Apache Commons Lang, Apache Derby Database Engine and Embedded JDBC Driver, Apache Geronimo, Apache Hadoop, Apache Hive, Apache HttpClient, Apache HttpComponents Client, Apache JAMES, Apache Log4j, Apache Lucene Core, Apache Neethi, Apache POI, Apache ServiceMix, Apache Tomcat, Apache Velocity, Apache WSS4J, Apache WebServices Common Utilities, Apache Xml-RPC, Apache Zookeeper, Box Java SDK (V2), CSV Tools, DataStax Java Driver for Apache Cassandra, Ehcache, Ezmorph, Ganymed SSH-2 for Java, Google APIs Client Library for Java, Google Gson, Groovy, Guava : Google Core Libraries for Java, H2 Embedded Database and JDBC Driver, Hector : A high level Java client for Apache Cassandra, Hibernate Validator, HighScale Lib, HsqlDB, Ini4j, JClouds, JLine, JSON, JSR 305 : Annotations for Software Defect Detection in Java, JUnit, Jackson Java JSON-processor, Java API for RESTful Services, Java Agent for Memory Measurements, Jaxb, Jaxen, Jettison, Jetty, Joda-Time, Json Simple, LightCouch, MetaStuff, Mondrian, OpenSAML, Paraccel JDBC Driver, PostgreSQL JDBC Driver, Resty : A simple HTTP REST client for Java, Rocoto, SL4J : Simple Logging Facade for Java, SQLite JDBC Driver, Simple API for CSS, SshJ, StAX API, StAXON - JSON via StAX, The Castor Project, The Legion of the Bouncy Castle, W3C, Woden, Woodstox : High-performance XML processor, Xalan-J, Xerces2, XmlBeans, XmlSchema Core, Xmlsec - Apache Santuario, Zip4J, atinject, dropbox-sdk-java : Java library for the Dropbox Core API, google-guice. Fournis sous leur licence respective.

Table des matières

Préface xxiii
Informations générales xxiii
Objectif xxiii
Public visé xxiii
Conventions typographiques xxiii
Remarques et Support xxiv
Composants Big Data 1
tGSBucketCreate 2
Propriétés du tGSBucketCreate 2
Scénario associé 3
tGSBucketDelete 4
Propriétés du tGSBucketDelete 4
Scénario associé 5
tGSBucketExist 6
Propriétés du tGSBucketExist 6
Scénario associé 7
tGSBucketList 8
Propriétés du tGSBucketList 8
Scénario associé 9
tGSClose 10
Propriétés du tGSClose 10
Scénario associé 10
tGSConnection 11
Propriétés du tGSConnection 11
Scénario associé 11
tGSCopy 12
Propriétés du tGSCopy 12
Scénario associé 13
tGSDelete 14
Propriétés du tGSDelete 14
Scénario associé 15
tGSGet 16
tGSGet properties 16
Scénario associé 17
tGSList 18
Propriétés du tGSList 18
Scénario associé 19
tGSPut 20
Propriétés du tGSPut 20
Scénario : Gérer des fichiers avec Google Cloud Storage 21
tHiveClose 28
Propriétés du tHiveClose 28
Scénario associé 29
tHiveConnection 30
Propriétés du tHiveConnection 30
Connexion à une distribution Hadoop personnalisée 36
Scénario : Créer une table Hive partitionnée 39
tHiveRow 45
Propriétés du tHiveRow 45
Scénarios associés 52
Composants Business 53
tAlfrescoOutput 54
Propriétés du tAlfrescoOutput 54
Scénario : Créer des documents sur un serveur Alfresco 59
tBonitaDeploy 65
Propriétés du tBonitaDeploy 65
Scénario associé 66
tBonitaInstantiateProcess 67
Propriétés du tBonitaInstantiateProcess 67
Scénario 1 : Exécuter un processus Bonita via un Job Talend 69
Scénario 2 : Ecrire en sortie l'UUID de l'instance du processus via le lien Row > Main 74
tCentricCRMInput 77
Propriétés du tCentricCRMInput 77
Scénario associé 78
tCentricCRMOutput 79
Propriétés du tCentricCRMOutput 79
Scénario associé 80
tHL7Input 81
Propriétés du tHL7Input 81
Scénario : Retrouver des informations concernant les patients et les événements à partir d'un fichier HL7 82
tHL7Output 86
Propriétés du tHL7Output 86
Scénario : Générer un accusé de réception (Acknowledgment) HL7 86
Scénario associé 93
tMarketoInput 94
Propriétés du tMarketoInput 94
Scénario associé 97
tMarketoListOperation 98
Propriétés du tMarketoListOperation 98
Scénario : Ajouter un enregistrement de prospects dans une liste de la base de données Marketo 99
tMarketoOutput 103
Propriétés du tMarketoOutput 103
Scénario : Insérer et récupérer des données d'une base de données Marketo via un système externe 105
tMicrosoftCrmInput 111
Propriétés du tMicrosoftCrmInput 111
Scénario : Ecrire des données dans une base de données Microsoft CRM et attribuer des conditions aux colonnes pour extraire des lignes spécifiques 113
tMicrosoftCrmOutput 120
Propriétés du tMicrosoftCrmOutput 120
Scénario associé 122
tMSAXInput 123
Propriétés du tMSAXInput 123
Scénario associé 124
tMSAXOutput 125
Propriétés du tMSAXOutput 125
Scénario : Insérer des données dans une table définie sur le serveur Microsoft AX 127
Scénario 2 : Effacer des données d'une table précise sur le serveur MicrosoftAX. 130
tNetsuiteInput 133
Propriétés du tNetsuiteInput 133
Scénario associé 134
tNetsuiteOutput 135
Propriétés du tNetsuiteOutput 135
Scénario associé 137
tOpenbravoERPInput 138
Propriétés du tOpenbravoERPInput 138
Scénario associé 139
tOpenbravoERPOutput 140
Propriétés du tOpenbravoERPOutput 140
Scénario associé 141
tSageX3Input 142
Propriétés du tSageX3Input 142
Scénario : Utiliser les clés des requêtes pour extraire des données d'un système Sage X3 donné 143
tSageX3Output 147
Propriétés du tSageX3Output 147

Scénario : Utiliser un service Web Sage X3 pour insérer des données dans un système Sage X3 donné 148
tSalesforceBulkExec 152
Propriétés du tSalesforceBulkExec 152
Scénario associé 154
tSalesforceConnection 155
Propriétés du tSalesforceConnection ... 155
Scénario : Utiliser OAuth2 pour vous connecter à Salesforce.com 157
tSalesforceGetDeleted 162
Propriétés du tSalesforceGetDeleted 162
Scénario : Récupérer les données supprimées du serveur Salesforce 164
tSalesforceGetServerTimestamp 167
Propriétés du tSalesforceGetServerTimestamp 167
Scénario associé 169
tSalesforceGetUpdated 170
Propriétés du tSalesforceGetUpdated ... 170
Scénario associé 172
tSalesforceInput 173
Propriétés du tSalesforceInput 173
Scénario : Utiliser des requêtes pour extraire des données d'une base de données Salesforce 176
tSalesforceOutput 180
Propriétés du tSalesforceOutput 180
Scénario 1 : Supprimer des données de l'objet Account 183
Scénario 2 : Rassembler des données en erreur tout en insérant des données dans un module sur Salesforce.com 185
Scénario 3 : Insérer des ID d'un fichier Excel vers le Module Contact ... 188
Scénario 4 : Effectuer un upsert du module Contact selon les relations de mapping avec les ID externes dans le module Account 192
tSalesforceOutputBulk 200
Propriétés du tSalesforceOutputBulk ... 200
Scénario : Insérer des données de masse transformées dans votre Salesforce.com 202
tSalesforceOutputBulkExec 207
Propriétés du tSalesforceOutputBulkExec 207
Scénario : Insérer des données de masse dans votre Salesforce.com 210
tSalesforceWaveBulkExec 214
Propriétés du tSalesforceWaveBulkExec 214
Scénario associé 216
tSalesforceWaveOutputBulkExec 217
Propriétés du tSalesforceWaveOutputBulkExec 217
Scénario associé 219
tSAPBWInput 220
Propriétés du tSAPBWInput 220
Scénario : Lire des données d'une base de données SAP BW 221
tSAPCommit 225
Propriétés du tSAPCommit 225
Scénario associé 225
tSAPConnection 226
Propriétés du tSAPConnection 226
Scénario associé 227
tSAPInput 228
Propriétés du tSAPInput 228
Scénario 1 : Récupérer les métadonnées d'un système SAP 230
Scénario 2 : Lire les données des différents schémas de la fonction RFC_READ_TABLE 237
Scénario 3 : Récupérer les données d'un système SAP 242
tSAPOutput 247
Propriétés du tSAPOutput 247
Scénarios associés 249
tSAPRollback 250
Propriétés du tSAPRollback 250
Scénario associé 250
tSugarCRMInput 251
Propriétés du tSugarCRMInput 251
Scénario: Extraire les données d'un compte à partir de SugarCRM 252
tSugarCRMOutput 254
Propriétés du tSugarCRMOutput 254
Scénario associé 255
tVtigerCRMInput 256
Propriétés du tVtigerCRMInput 256
Scénario associé 257
tVtigerCRMOutput 258
Propriétés du tVtigerCRMOutput 258
Scénario associé 259
Composants Business Intelligence 261
tBarChart 262
Propriétés du tBarChart 262
Scénario : Créer un diagramme en barres à partir de données d'entrée 263
tDB2SCD 270
Propriétés du tDB2SCD 270
Scénario associé 272
tDB2SCDELT 273
Propriétés du tDB2SCDELT 273
Scénario associé 275
tGreenplumSCD 276
Propriétés du tGreenplumSCD 276
Scénario associé 278
tInformixSCD 279
Propriétés du tInformixSCD 279
Scénario associé 281
tIngresSCD 282
Propriétés du tIngresSCD 282
Scénario associé 283
tJasperOutput 284
Propriétés du tJasperOutput 284
Scénario : Générer un rapport avec un modèle .jrxml 285
tJasperOutputExec 288
Propriétés du tJasperOutputExec 288
Scénario associé 289
tLineChart 290
Propriétés du tLineChart 290
Scénario : Créer un graphique en lignes afin de faciliter l'analyse des tendances 291
tMondrianInput 299
Propriétés du tMondrianInput 299
Scénario : Tables en jointure croisée ... 300
tMSSqlSCD 304
Propriétés du tMSSqlSCD 304
Scénario associé 306
tMysqlSCD 307
Propriétés du tMysqlSCD 307
Scénario : Traquer des modifications avec les Slowly Changing Dimensions de type 0 à 3 311
tMysqlSCDELT 319
Propriétés du tMysqlSCDELT 319
Scénario associé 321
tNetezzaSCD 322
Propriétés du tNetezzaSCD 322

Scénario associé 324
tOracleSCD 325
Propriétés du tOracleSCD 325
Scénario associé 327
tOracleSCDELT 328
Propriétés du tOracleSCDELT 328
Scénario associé 330
tPaloCheckElements 331
Propriétés du tPaloCheckElements 331
Scénario associé 333
tPaloClose 334
Propriétés du tPaloClose 334
Scénario associé 334
tPaloConnection 335
Propriétés du tPaloConnection 335
Scénario associé 336
tPaloCube 337
Propriétés du tPaloCube 337
Scénario : Créer un cube dans une base de données existante 338
tPaloCubeList 341
Propriétés du tPaloCubeList 341
Découvrir le schéma de sortie en lecture seule du tPaloCubeList 342
Scénario : Récupérer des informations détaillées d'un cube d'une base de données déterminée 343
tPaloDatabase 345
Propriétés du tPaloDatabase 345
Scénario : Créer une base de données 346
tPaloDatabaseList 348
Propriétés du tPaloDatabaseList 348
Découvrir le schéma de sortie en lecture seule du tPaloDatabaseList 349
Scénario : Récupérer des informations détaillées concernant les bases de données d'un serveur Palo donné 349
tPaloDimension 352
Propriétés du tPaloDimension 352
Scénario : Créer une dimension avec des éléments 355
tPaloDimensionList 361
Propriétés du tPaloDimensionList 361
Découvrir le schéma de sortie en lecture seule du tPaloDimensionList 363
Scénario: Récupérer des informations détaillées concernant les dimensions d'une base de données déterminée 363
tPaloInputMulti 366
Propriétés du tPaloInputMulti 366
Scénario : Récupérer des éléments de dimensions dans un cube donné 368
tPaloOutput 372
Propriétés du tPaloOutput 372
Scénario associé 373
tPaloOutputMulti 374
Propriétés du tPaloOutputMulti 374
Scénario 1 : Ecrire des données dans un cube donné 376
Scénario 2 : Refuser un flux de données d'entrée lorsque les éléments à écrire n'existent pas dans un cube donné 379
tPaloRule 383
Propriétés du tPaloRule 383
Scénario : Créer une règle dans un cube donné 384
tPaloRuleList 387
Propriétés du tPaloRuleList 387
Découvrir le schéma de sortie en lecture seule du tPaloRuleList 389
Scénario : Récupérer des informations détaillées concernant des règles d'un cube donné 389
tParAccelSCD 391
Propriétés du tParAccelSCD 391
Scénario associé 393
tPostgresPlusSCD 394
Propriétés du tPostgresPlusSCD 394
Scénario associé 396
tPostgresPlusSCDELT 397
Propriétés du tPostgresPlusSCDELT 397
Scénario associé 399
tPostgresqlSCD 400
Propriétés du tPostgresqlSCD 400
Scénario associé 402
tPostgresqlSCDELT 403
Propriétés du tPostgresqlSCDELT 403
Scénario associé 405
tSPSSInput 406
Propriétés du tSPSSInput 406
Scénario : Afficher le contenu d'un fichier SPSS 407
tSPSSOutput 410
Propriétés du tSPSSOutput 410
Scénario : Ecrire des données dans un fichier .sav 411
tSPSSProperties 414
Propriétés du tSPSSProperties 414
Scénario associé 415
tSPSSStructure 416
Propriétés du tSPSSStructure 416
Scénario associé 417
tSybaseSCD 418
Propriétés du tSybaseSCD 418
Scénario associé 420
tSybaseSCDELT 421
Propriétés du tSybaseSCDELT 421
Scénario associé 423
tVerticaSCD 424
Propriétés du tVerticaSCD 424
Conseils et techniques 426
Scénario associé 426
Composants Cloud 427
tAmazonAuroraClose 428
Propriétés du tAmazonAuroraClose 428
Scénario associé 428
tAmazonAuroraCommit 430
Propriétés du tAmazonAuroraCommit 430
Scénario associé 431
tAmazonAuroraConnection 432
Propriétés du tAmazonAuroraConnection 432
Scénario associé 433
tAmazonAuroraInput 434
Propriétés du tAmazonAuroraInput 434
Scénarios associés 436
tAmazonAuroraOutput 437
Propriétés du tAmazonAuroraOutput 437
Scénarios associés 441
tAmazonAuroraRollback 442
Propriétés du tAmazonAuroraRollback 442
Scénario associé 442
tAmazonMysqlClose 444
Propriétés du tAmazonMysqlClose 444
Scénario associé 444
tAmazonMysqlCommit 445
Propriétés du tAmazonMysqlCommit 445

Scénario associé 446
tAmazonMysqlConnection 447
Propriétés du tAmazonMysqlConnection 447
Scénario associé 448
tAmazonMysqlInput 449
Propriétés du tAmazonMysqlInput 449
Scénarios associés 451
tAmazonMysqlOutput 452
Propriétés du tAmazonMysqlOutput 452
Scénarios associés 455
tAmazonMysqlRollback 456
Propriétés du tAmazonMysqlRollback 456
Scénario associé 456
tAmazonMysqlRow 457
Propriétés du tAmazonMysqlRow 457
Scénario associé 459
tAmazonOracleClose 460
Propriétés du tAmazonOracleClose 460
Scénario associé 460
tAmazonOracleCommit 461
Propriétés du tAmazonOracleCommit 461
Scénario associé 462
tAmazonOracleConnection 463
Propriétés du tAmazonOracleConnection 463
Scénario associé 464
tAmazonOracleInput 465
Propriétés du tAmazonOracleInput 465
Scénarios associés 467
tAmazonOracleOutput 468
Propriétés du tAmazonOracleOutput ... 468
Scénarios associés 471
tAmazonOracleRollback 472
Propriétés du tAmazonOracleRollback 472
Scénario associé 472
tAmazonOracleRow 473
Propriétés du tAmazonOracleRow 473
Scénarios associés 475
tBoxConnection 476
Propriétés du tBoxConnection 476
Scénario associé 477
tBoxCopy 478
Propriétés du tBoxCopy 478
Scénario associé 479
tBoxDelete 480
Propriétés du tBoxDelete 480
Scénario associé 481
tBoxGet 482
Propriétés du tBoxGet 482
Scénario associé 483
tBoxList 484
Propriétés du tBoxList 484
Scénario associé 485
tBoxPut 486
Propriétés du tBoxPut 486
Scénario : Charger et télécharger des fichiers depuis Box 487
tCloudStart 491
Propriétés du tCloudStart 491
Scénario associé 492
tCloudStop 493
Propriétés du tCloudStop 493
Scénario associé 494
tDropboxConnection 495
Propriétés du tDropboxConnection 495
Scénario associé 495
tDropboxDelete 496
Propriétés du tDropboxDelete 496
Scénario associé 496
tDropboxGet 497
Propriétés du tDropboxGet 497
Scénario associé 498
tDropboxList 499
Propriétés du tDropboxList 499
Scénario associé 500
tDropboxPut 501
Propriétés du tDropboxPut 501
Scénario : Télécharger des fichiers dans Dropbox 502
tGoogleDriveConnection 508
Propriétés du tGoogleDriveConnection 508
Obtenir un jeton d'accès à Google Drive 509
Scénario associé 514
tGoogleDriveCopy 515
Propriétés du tGoogleDriveCopy 515
Scénario associé 516
tGoogleDriveCreate 517
Propriétés du tGoogleDriveCreate 517
Scénario associé 518
tGoogleDriveDelete 519
Propriétés du tGoogleDriveDelete 519
Scénario associé 520
tGoogleDriveGet 521
Propriétés du tGoogleDriveGet 521
Scénario associé 522
tGoogleDriveList 523
Propriétés du tGoogleDriveList 523
Scénario associé 524
tGoogleDrivePut 525
Propriétés du tGoogleDrivePut 525
Gestion des fichiers avec Google Drive 526
tGSBucketCreate 535
tGSBucketDelete 536
tGSBucketExist 537
tGSBucketList 538
tGSClose 539
tGSConnection 540
tGSCopy 541
tGSDelete 542
tGSGet 543
tGSList 544
tGSPut 545
tMarketoInput 546
tMarketoListOperation 547
tMarketoOutput 548
tNetsuiteInput 549
tNetsuiteOutput 550
tS3BucketCreate 551
Propriétés du tS3BucketCreate 551
Scénario associé 552
tS3BucketDelete 553
tS3BucketDelete properties 553
Scénario associé 554
tS3BucketExist 555
Propriétés du tS3BucketExist 555
Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3 556
tS3BucketList 561
Propriétés du tS3BucketList 561
Scénario associé 562
tS3Close 563
Propriétés du tS3Close 563
Scénario associé 564
tS3Connection 565
Propriétés du tS3Connection 565
Scénario associé 566

tS3Delete **567**
Propriétés du tS3Delete 567
Scénario associé 568
tS3Get **569**
Propriétés du tS3Get 569
Scénario associé 570
tS3List **571**
Propriétés du tS3List 571
Scénario : Lister les fichiers d'un bucket ayant le même préfixe 572
tS3Put **577**
Propriétés du tS3Put 577
Scénario : Echange de fichiers avec Amazon S3 578
tSalesforceBulkExec **581**
tSalesforceConnection **582**
tSalesforceGetDeleted **583**
tSalesforceGetServerTimestamp **584**
tSalesforceGetUpdated **585**
tSalesforceInput **586**
tSalesforceOutput **587**
tSalesforceOutputBulk **588**
tSalesforceOutputBulkExec **589**
tSalesforceWaveBulkExec **590**
tSalesforceWaveOutputBulkExec **591**
tSugarCRMInput **592**
tSugarCRMOutput **593**
Composants Custom Code **595**
tGroovy **596**
Propriétés du tGroovy 596
Scénarios associés 596
tGroovyFile **597**
Propriétés du tGroovyFile 597
Scénario : Appeler du code Groovy contenu dans un fichier 597
tJava **599**
Propriétés du tJava 599
Scénario : Imprimer le contenu d'une variable 599
tJavaFlex **603**
Propriétés du tJavaFlex 603
Scénario 1 : Générer un flux de données 604
Scénario 2 : Traiter des lignes de données avec le tJavaFlex 607
tJavaRow **610**
Propriétés du tJavaRow 610
Scénario : Transformer des données ligne par ligne avec un tJavaRow 611
tLibraryLoad **615**
Propriétés du tLibraryLoad 615
Scénario : Vérifier le format d'une adresse e-mail 616
tSetGlobalVar **618**
Propriétés du tSetGlobalVar 618
Scénario : Afficher le contenu d'une variable globale 618
Composants Data Quality **621**
tAddCRCRow **622**
Propriétés du tAddCRCRow 622
Scénario : Ajouter une clé de substitution à un fichier 623
tChangeFileEncoding **625**
tExtractRegexFields **626**
tFuzzyMatch **627**
Propriétés du tFuzzyMatch 627
Scénario 1 : Distance de Levenshtein de 0 pour les prénoms 628
Scénario 2 : Distance de Levenshtein de 1 ou 2 pour les prénoms 631
Scénario 3 : Distance métaphonique pour les prénoms 632
tIntervalMatch **633**
Propriétés du tIntervalMatch 633
Scénario : Identifier l'emplacement de serveurs à partir de leur IP 634
tReplaceList **638**
Propriétés du tReplaceList 638
Scénario : Remplacement à partir d'un fichier de référence 639
tSchemaComplianceCheck **643**
Propriétés du tSchemaComplianceCheck 643
Scénario : Valider les dates en fonction d'un schéma 645
tUniqRow **649**
Propriétés du tUniqRow 649
Scénario 1 : Dédoublonner des données 650
Scénario 2 : Dédoublonner des entrées en utilisant des composants Map/Reduce 653
tUniservBTGeneric **665**
Propriétés du tUniservBTGeneric 665
Scénario : Exécution d'un processus dans *DQ Batch Suite* 667
tUniservRTConvertName **671**
Propriétés du tUniservRTConvertName 671
Scénario : Analyse d'une ligne de nom et attribution d'un titre 672
tUniservRTMailBulk **676**
Propriétés du tUniservRTMailBulk 676
Scénario : Créer un pool d'index 676
tUniservRTMailOutput **680**
Propriétés du tUniservRTMailOutput 680
Scénario associé 681
tUniservRTMailSearch **682**
Propriétés du tUniservRTMailSearch ... 682
Scénario : Ajouter des contacts au pool d'index de mailRetrieval 683
tUniservRTPost **688**
Propriétés du tUniservRTPost 688
Scénario 1 : Vérifier et corriger le code postal, la ville et la rue 689
Scénario 2 : Vérifier et corriger le code postal, la ville et la rue, tout en rejetant les données incorrectes 693
Databases - Composants traditionnels **695**
tAccessBulkExec **696**
Propriétés du tAccessBulkExec 696
Scénarios associés 698
tAccessClose **699**
Propriétés du tAccessClose 699
Scénario associé 699
tAccessCommit **700**
Propriétés du tAccessCommit 700
Scénario associé 700
tAccessConnection **701**
Propriétés du tAccessConnection 701
Scénario: Insérer des données dans des tables parent/enfant 702
tAccessInput **706**
Propriétés du tAccessInput 706
Scénarios associés 708
tAccessOutput **709**
Propriétés du tAccessOutput 709
Scénarios associés 712
tAccessOutputBulk **713**
Propriétés du tAccessOutputBulk 713
Scénarios associés 714
tAccessOutputBulkExec **715**

Propriétés du tAccessOutputBulkExec 715
Scénarios associés 717
tAccessRollback 718
Propriétés du tAccessRollback 718
Scénario associé 718
tAccessRow 719
Propriétés du tAccessRow 719
Scénarios associés 721
tAS400Close 722
Propriétés du tAS400Close 722
Scénario associé 722
tAS400Commit 723
Propriétés du tAS400Commit 723
Scénario associé 723
tAS400Connection 724
Propriétés du tAS400Connection 724
Scénario associé 725
tAS400Input 726
Propriétés du tAS400Input 726
Scénarios associés 727
tAS400LastInsertId 728
Propriétés du tAS400LastInsertId 728
Scénario associé 729
tAS400Output 730
Propriétés du tAS400Output 730
Scénarios associés 732
tAS400Rollback 734
Propriétés du tAS400Rollback 734
Scénario associé 734
tAS400Row 735
Propriétés du tAS400Row 735
Scénarios associés 737
tDB2BulkExec 738
Propriétés du tDB2BulkExec 738
Scénario associé 740
tDB2Close 741
Propriétés du tDB2Close 741
Scénario associé 741
tDB2Commit 742
Propriétés du tDB2Commit 742
Scénario associé 742
tDB2Connection 743
Propriétés du tDB2Connection 743
Scénarios associés 744
tDB2Input 745
Propriétés du tDB2Input 745
Scénarios associés 747
tDB2Output 748
Propriétés du tDB2Output 748
Scénarios associés 751
tDB2Rollback 752
Propriétés du tDB2Rollback 752
Scénario associé 752
tDB2Row 753
Propriétés du tDB2Row 753
Scénarios associés 755
tDB2SCD 756
tDB2SCDELT 757
tDB2SP 758
Propriétés du tDB2SP 758
Scénario associé 760
tInformixBulkExec 761
Propriétés du tInformixBulkExec 761
Scénario associé 764
tInformixClose 765
Propriétés du tInformixClose 765
Scénario associé 765
tInformixCommit 766
Propriétés du tInformixCommit 766
Scénario associé 767
tInformixConnection 768
Propriétés du tInformixConnection 768
Scénarios associés 769
tInformixInput 770
Propriétés du tInformixInput 770
Scénarios associés 772
tInformixOutput 773
Propriétés du tInformixOutput 773
Scénarios associés 776
tInformixOutputBulk 777
Propriétés du tInformixOutputBulk 777
Scénarios associés 779
tInformixOutputBulkExec 780
Propriétés du tInformixOutputBulkExec 780
Scénarios associés 782
tInformixRollback 783
Propriétés du tInformixRollback 783
Scénario associé 783
tInformixRow 784
Propriétés du tInformixRow 784
Scénarios associés 786
tInformixSCD 787
tInformixSP 788
Propriétés du tInformixSP 788
Scénarios associés 790
tMSSqlBulkExec 791
Propriétés du tMSSqlBulkExec 791
Scénarios associés 793
tMSSqlColumnList 794
Propriétés du tMSSqlColumnList 794
Scénario associé 795
tMSSqlClose 796
Propriétés du tMSSqlClose 796
Scénario associé 796
tMSSqlCommit 797
Propriétés du tMSSqlCommit 797
Scénario associé 797
tMSSqlConnection 798
Propriétés du tMSSqlConnection 798
Scénario : Insérer des données dans une table de base de données et en extraire des informations 799
tMSSqlInput 805
Propriétés du tMSSqlInput 805
Scénarios associés 807
tMSSqlLastInsertId 808
Propriétés du tMSSqlLastInsertId 808
Scénario associé 809
tMSSqlOutput 810
Propriétés du tMSSqlOutput 810
Scénarios associés 814
tMSSqlOutputBulk 815
Propriétés du tMSSqlOutputBulk 815
Scénarios associés 816
tMSSqlOutputBulkExec 817
Propriétés du tMSSqlOutputBulkExec 817
Scénarios associés 819
tMSSqlRollback 820
Propriétés du tMSSqlRollback 820
Scénario associé 820
tMSSqlRow 821
Propriétés du tMSSqlRow 821
Scénarios associés 823
tMSSqlSCD 824
tMSSqlSP 825
Propriétés du tMSSqlSP 825
Scénario : Récupérer des informations personnelles à l'aide d'une procédure stockée 827
Scénarios associés 831
tMSSqlTableList 832

Propriétés du tMSSqlTableList 832
Scénario associé 833
tMysqlBulkExec 834
Propriétés du tMysqlBulkExec 834
Scénarios associés 836
tMysqlClose 837
Propriétés du tMysqlClose 837
Scénario associé 837
tMysqlColumnList 838
Propriétés du tMysqlColumnList 838
Scénario : Itérer une table de base de données et lister le nom des colonnes de la table 839
tMysqlCommit 842
Propriétés du tMysqlCommit 842
Scénario associé 843
tMysqlConnection 844
Propriétés du tMysqlConnection 844
Scénario 1 : Insérer des données dans des tables mère/fille 845
Scénario 2 : Partager une connexion de base de données entre un Job père et un Job fils 848
tMysqlInput 855
Propriétés du tMysqlInput 855
Scénario 1 : Ecrire des colonnes d'une base de données MySQL dans un fichier de sortie 857
Scénario 2 : Utiliser des paramètres de contexte lors de la lecture d'une table d'une base de données MySQL ... 860
Scénario 3 : Lire des données dans des bases de données MySQL à l'aide de connexions dynamiques basées sur les variables de contexte 863
tMysqlLastInsertId 870
Propriétés du tMysqlLastInsertId 870
Scénario : Récupérer les ID des dernières entrées ajoutées 871
tMysqlOutput 876
Propriétés du tMysqlOutput 876
Scénario 1 : Ajouter une colonne et modifier les données 879
Scénario 2 : Effectuer des mises à jour dans une base de données 884
Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject 887
tMysqlOutputBulk 893
Propriétés du tMysqlOutputBulk 893
Scénario : Insérer des données transformées dans une base MySQL 894
tMysqlOutputBulkExec 899
Propriétés du tMysqlOutputBulkExec 899
Scénario : Insérer des données dans une base MySQL 901
tMysqlRollback 903
Propriétés du tMysqlRollback 903
Scénario : Annuler l'insertion de données dans des tables mère/fille 903
tMysqlRow 905
Propriétés du tMysqlRow 905
Scénario 1 : Supprimer et re-générer un index de table MySQL 907
Scénario 2 : Utiliser l'instance PreparedStatement pour faire une requête sur des données 909
Scénario 3 : Combiner deux flux pour une sortie sélective 914
tMysqlSCD 920
tMysqlSCDELT 921
tMysqlSP 922
Propriétés du tMysqlSP 922
Scénario : Trouver le libellé State à l'aide d'une procédure stockée 923
Scénarios associés 926
tMysqlTableList 927
Propriétés du tMysqlTableList 927
Scénario : Effectuer une opération d'itération sur des tables et en effacer le contenu grâce à un modèle SQL défini par l'utilisateur (SQL Template) 928
Scénario associé 932
tOleDbInput 933
Propriétés du tOleDbInput 933
Scénario associé 934
tOleDbOutput 935
Propriétés du tOleDbOutput 935
Scénario associé 937
tOleDbRow 938
Propriétés du tOleDbRow 938
Scénario associé 939
tOracleBulkExec 940
Propriétés du tOracleBulkExec 940
Scénario : Supprimer et insérer des données dans une base Oracle 943
tOracleClose 947
Propriétés du tOracleClose 947
Scénario associé 947
tOracleCommit 948
Propriétés du tOracleCommit 948
Scénario associé 948
tOracleConnection 949
Propriétés du tOracleConnection 949
Scénario associé 950
tOracleInput 951
Propriétés du tOracleInput 951
Scénario : Utiliser les paramètres de contexte lors de la lecture d'une table d'une base de données Oracle 953
Scénarios associés 957
tOracleOutput 958
Propriétés du tOracleOutput 958
Scénarios associés 962
tOracleOutputBulk 963
Propriétés du tOracleOutputBulk 963
Scénarios associés 964
tOracleOutputBulkExec 966
Propriétés du tOracleOutputBulkExec 966
Scénarios associés 969
tOracleRollback 970
Propriétés du tOracleRollback 970
Scénario associé 970
tOracleRow 971
Propriétés du tOracleRow 971
Scénario associé 974
tOracleSCD 975
tOracleSCDELT 976
tOracleSP 977
Propriétés du tOracleSP 977
Scénario : Vérifier le format de numéros à l'aide d'une procédure stockée 979
tOracleTableList 984
Propriétés du tOracleTableList 984
Scénario associé 985
tPostgresqlBulkExec 986
Propriétés du tPostgresqlBulkExec 986
Scénarios associés 988
tPostgresqlCommit 989
Propriétés du tPostgresqlCommit 989
Scénario associé 989

tPostgresqlClose **990**
Propriétés du tPostgresqlClose 990
Scénario associé 990
tPostgresqlConnection **991**
Propriétés du tPostgresqlConnection ... 991
Scénario associé 992
tPostgresqlInput **993**
Propriétés du tPostgresqlInput 993
Scénarios associés 995
tPostgresqlOutput **996**
Propriétés du tPostgresqlOutput 996
Scénarios associés 999
tPostgresqlOutputBulk **1000**
Propriétés du tPostgresqlOutputBulk .. 1000
Scénarios associés 1001
tPostgresqlOutputBulkExec **1002**
Propriétés du tPostgresqlOutputBulkExec 1002
Scénarios associés 1004
tPostgresqlRollback **1005**
Propriétés du tPostgresqlRollback 1005
Scénario associé 1005
tPostgresqlRow **1006**
Propriétés du tPostgresqlRow 1006
Scénarios associés 1008
tPostgresqlSCD **1009**
tPostgresqlSCDELT **1010**
tSybaseBulkExec **1011**
Propriétés du tSybaseBulkExec 1011
Scénarios associés 1013
tSybaseClose **1015**
Propriétés du tSybaseClose 1015
Scénario associé 1015
tSybaseCommit **1016**
Propriétés du tSybaseCommit 1016
Scénario associé 1016
tSybaseConnection **1017**
Propriétés du tSybaseConnection 1017
Scénarios associés 1018
tSybaseInput **1019**
Propriétés du tSybaseInput 1019
Scénario associé 1021
tSybaseIQBulkExec **1022**
Propriétés du tSybaseIQBulkExec 1022
Scénario associé 1024
tSybaseIQOutputBulkExec **1025**
Propriétés du tSybaseIQOutputBulkExec 1025
Scénario : Charger en masse de données dans une base de données Sybase IQ 12 1027
Scénarios associés 1030
tSybaseOutput **1031**
tSybaseOutput Properties 1031
Scénario associé 1034
tSybaseOutputBulk **1035**
Propriétés du tSybaseOutputBulk 1035
Scénario associé 1036
tSybaseOutputBulkExec **1037**
Propriétés du tSybaseOutputBulkExec 1037
Scénarios associés 1039
tSybaseRollback **1040**
Propriétés du tSybaseRollback 1040
Scénario associé 1040
tSybaseRow **1041**
Propriétés du tSybaseRow 1041
Scénario associé 1043
tSybaseSCD **1044**
tSybaseSCDELT **1045**
tSybaseSP **1046**
Propriétés du tSybaseSP 1046
Scénario associé 1048
tVerticaSCD **1049**

Databases - Composants d'appliance et de datawarehouse 1051

tGreenplumBulkExec **1052**
Propriétés du tGreenplumBulkExec ... 1052
Scénarios associés 1054
tGreenplumClose **1055**
Propriétés du tGreenplumClose 1055
Scénario associé 1055
tGreenplumCommit **1056**
Propriétés du tGreenplumCommit 1056
Scénario associé 1056
tGreenplumConnection **1058**
Propriétés du tGreenplumConnection 1058
Scénarios associés 1059
tGreenplumGPLoad **1060**
Propriétés du tGreenplumGPLoad 1060
Scénario associé 1064
tGreenplumInput **1065**
Propriétés du tGreenplumInput 1065
Scénarios associés 1066
tGreenplumOutput **1068**
Propriétés du tGreenplumOutput 1068
Scénarios associés 1070
tGreenplumOutputBulk **1072**
Propriétés du tGreenplumOutputBulk 1072
Scénarios associés 1073
tGreenplumOutputBulkExec **1074**
Propriétés du tGreenplumOutputBulkExec 1074
Scénarios associés 1076
tGreenplumRollback **1077**
Propriétés du tGreenplumRollback 1077
Scénario associé 1077
tGreenplumRow **1078**
Propriétés du tGreenplumRow 1078
Scénarios associés 1080
tGreenplumSCD **1081**
tIngresBulkExec **1082**
Propriétés du tIngresBulkExec 1082
Scénario associé 1084
tIngresClose **1085**
Propriétés du tIngresClose 1085
Scénario associé 1085
tIngresCommit **1086**
Propriétés du tIngresCommit 1086
Scénario associé 1086
tIngresConnection **1087**
Propriétés du tIngresConnection 1087
Scénario associé 1088
tIngresInput **1089**
Propriétés du tIngresInput 1089
Scénarios associés 1090
tIngresOutput **1091**
Propriétés du tIngresOutput 1091
Scénarios associés 1093
tIngresOutputBulk **1095**
Propriétés du tIngresOutputBulk 1095
Scénario associé 1096
tIngresOutputBulkExec **1097**
tIngresOutputBulkExec 1097
Scénario : Charger des données dans une table du SGBD Ingres 1099
Scénarios associés 1102
tIngresRollback **1103**
Propriétés du tIngresRollback 1103
Scénario associé 1103
tIngresRow **1104**
Propriétés du tIngresRow 1104

Scénarios associés 1106
tIngresSCD 1107
tNetezzaBulkExec 1108
Propriétés du tNetezzaBulkExec 1108
Scénarios associés 1110
tNetezzaClose 1111
Propriétés du tNetezzaClose 1111
Scénario associé 1111
tNetezzaCommit 1112
Propriétés du tNetezzaCommit 1112
Scénario associé 1112
tNetezzaConnection 1113
Propriétés du tNetezzaConnection 1113
Scénario associé 1114
tNetezzaInput 1115
Propriétés du tNetezzaInput 1115
Scénarios associés 1117
tNetezzaNzLoad 1118
Propriétés du tNetezzaNzLoad 1118
Scénario associé 1123
tNetezzaOutput 1124
Propriétés du tNetezzaOutput 1124
Scénarios associés 1127
tNetezzaRollback 1128
Propriétés du tNetezzaRollback 1128
Scénario associé 1128
tNetezzaRow 1129
Propriétés du tNetezzaRow 1129
Scénario associé 1131
tNetezzaSCD 1132
tParAccelBulkExec 1133
Propriétés du tParAccelBulkExec 1133
Scénarios associés 1135
tParAccelClose 1136
Propriétés du tParAccelClose 1136
Scénario associé 1136
tParAccelCommit 1137
Propriétés du tParAccelCommit 1137
Scénario associé 1137
tParAccelConnection 1138
Propriétés du tParAccelConnection ... 1138
Scénario associé 1139
tParAccelInput 1140
Propriétés du tParAccelInput 1140
Scénarios associés 1142
tParAccelOutput 1143
Propriétés du tParAccelOutput 1143
Scénarios associés 1146
tParAccelOutputBulk 1147
Propriétés du tParAccelOutputBulk ... 1147
Scénarios associés 1148
tParAccelOutputBulkExec 1149
Propriétés du tParAccelOutputBulkExec 1149
Scénarios associés 1151
tParAccelRollback 1152
Propriétés du tParAccelRollback 1152
Scénario associé 1152
tParAccelRow 1153
Propriétés du tParAccelRow 1153
Scénarios associés 1155
tParAccelSCD 1156
tRedshiftBulkExec 1157
Propriétés du tRedshiftBulkExec 1157
Charger/retirer des données dans/de Amazon S3 1160
tRedshiftClose 1170
Propriétés du tRedshiftClose 1170
Scénario associé 1170
tRedshiftCommit 1171
Propriétés du tRedshiftCommit 1171
Scénario associé 1171
tRedshiftConnection 1172
Propriétés du tRedshiftConnection 1172
Scénario associé 1173
tRedshiftInput 1174
Propriétés du tRedshiftInput 1174
Scénario : Manipuler des données avec Redshift 1176
tRedshiftOutput 1181
Propriétés du tRedshiftOutput 1181
Scénarios associés 1184
tRedshiftOutputBulk 1185
Propriétés du tRedshiftOutputBulk 1185
Scénario associé 1186
tRedshiftOutputBulkExec 1187
Propriétés du tRedshiftOutputBulkExec 1187
Scénario associé 1190
tRedshiftRollback 1191
Propriétés du tRedshiftRollback 1191
Scénario associé 1191
tRedshiftRow 1192
Propriétés du tRedshiftRow 1192
Scénarios associés 1194
tRedshiftUnload 1195
Propriétés du tRedshiftUnload 1195
Scénario associé 1197
tTeradataClose 1198
Propriétés du tTeradataClose 1198
Scénario associé 1198
tTeradataCommit 1199
Propriétés du tTeradataCommit 1199
Scénario associé 1199
tTeradataConnection 1200
Propriétés du tTeradataConnection 1200
Scénario associé 1201
tTeradataFastExport 1202
Propriétés du tTeradataFastExport 1202
Scénario associé 1203
tTeradataFastLoad 1204
Propriétés du tTeradataFastLoad 1204
Scénario associé 1205
tTeradataFastLoadUtility 1206
Propriétés du tTeradataFastLoadUtility 1206
Scénario associé 1207
tTeradataInput 1208
Propriétés du tTeradataInput 1208
Scénario associé 1210
tTeradataMultiLoad 1211
Propriétés du tTeradataMultiLoad 1211
Scénario associé 1213
tTeradataOutput 1214
Propriétés du tTeradataOutput 1214
Scénario associé 1217
tTeradataRollback 1218
Propriétés du tTeradataRollback 1218
Scénario associé 1218
tTeradataRow 1219
Propriétés du tTeradataRow 1219
Scénario associé 1221
tTeradataTPTExec 1222
Propriétés du tTeradataTPTExec 1222
Scénario : Charger des données dans une base de données Teradata 1226
tTeradataTPTUtility 1233
Propriétés du tTeradataTPTUtility 1233
Scénario associé 1236
tTeradataTPump 1237
Propriétés du tTeradataTPump 1237
Scénario : Insérer des données dans une table d'une base de données Teradata 1239

tVectorWiseCommit **1243**
Propriétés du tVectorWiseCommit 1243
Scénario associé 1243
tVectorWiseConnection **1244**
Propriétés du tVectorWiseConnection 1244
Scénario associé 1245
tVectorWiseInput **1246**
Propriétés du tVectorWiseInput 1246
Scénario associé 1248
tVectorWiseOutput **1249**
Propriétés du tVectorWiseOutput 1249
Scénario associé 1252
tVectorWiseRollback **1253**
Propriétés du tVectorWiseRollback ... 1253
Scénario associé 1253
tVectorWiseRow **1254**
Propriétés du tVectorWiseRow 1254
Scénario associé 1256
tVerticaBulkExec **1257**
Propriétés du tVerticaBulkExec 1257
Conseils et techniques 1259
Scénarios associés 1259
tVerticaClose **1260**
Propriétés du tVerticaClose 1260
Conseils et techniques 1260
Scénario associé 1260
tVerticaCommit **1261**
Propriétés du tVerticaCommit 1261
Conseils et techniques 1261
Scénario associé 1262
tVerticaConnection **1263**
Propriétés du tVerticaConnection 1263
Conseils et techniques 1264
Scénario associé 1264
tVerticaInput **1265**
Propriétés du tVerticaInput 1265
Conseils et techniques 1267
Scénarios associés 1267
tVerticaOutput **1268**
Propriétés du tVerticaOutput 1268
Conseils et techniques 1272
Scénarios associés 1272
tVerticaOutputBulk **1273**
Propriétés du tVerticaOutputBulk 1273
Conseils et techniques 1274
Scénarios associés 1274
tVerticaOutputBulkExec **1275**
Propriétés du tVerticaOutputBulkExec 1275
Conseils et techniques 1277
Scénarios associés 1277
tVerticaRollback **1278**
Propriétés du tVerticaRollback 1278
Conseils et techniques 1278
Scénario associé 1278
tVerticaRow **1279**
Propriétés du tVerticaRow 1279
Conseils et techniques 1281
Scénario associé 1281
Databases - Autres composants **1283**
tCreateTable **1284**
Propriétés du tCreateTable 1284
Scénario : Créer une nouvelle table dans une base de données MySQL 1287
tDBInput **1289**
Propriétés du tDBInput 1289
Scénario 1 : Afficher les données sélectionnées à partir d'une table 1291
Scénario 2 : Utiliser la variable StoreSQLQuery 1292
tDBOutput **1294**
Propriétés du tDBOutput 1294
Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC 1297
tDBSQLRow **1299**
Propriétés du tDBSQLRow 1299
Scénario : Réinitialiser l'auto-incrémentation d'une base de données 1301
tEXAInput **1303**
Propriétés du tEXAInput 1303
Scénarios associés 1304
tEXAOutput **1305**
Propriétés du tEXAOutput 1305
Scénarios associés 1308
tEXARow **1309**
Propriétés du tEXARow 1309
Scénarios associés 1311
tEXistConnection **1312**
Propriétés du tEXistConnection 1312
Scénario associé 1312
tEXistDelete **1313**
Propriétés du tEXistDelete 1313
Scénario associé 1314
tEXistGet **1315**
Propriétés du tEXistGet 1315
Scénario : Récupérer des ressources à partir d'un serveur distant de base de données eXist 1316
tEXistList **1319**
Propriétés du tEXistList 1319
Scénario associé 1320
tEXistPut **1321**
Propriétés du tEXistPut 1321
Scénario associé 1322
tEXistXQuery **1323**
Propriétés du tEXistXQuery 1323
Scénario associé 1324
tEXistXUpdate **1325**
Propriétés du tEXistXUpdate 1325
Scénario associé 1326
tFirebirdClose **1327**
Propriétés du tFirebirdClose 1327
Scénario associé 1327
tFirebirdCommit **1328**
Propriétés du tFirebirdCommit 1328
Scénario associé 1329
tFirebirdConnection **1330**
Propriétés du tFirebirdConnection 1330
Scénario associé 1331
tFirebirdInput **1332**
Propriétés du tFirebirdInput 1332
Scénarios associés 1333
tFirebirdOutput **1335**
Propriétés du tFirebirdOutput 1335
Scénarios associés 1338
tFirebirdRollback **1339**
Propriétés du tFirebirdRollback 1339
Scénario associé 1339
tFirebirdRow **1340**
Propriétés du tFirebirdRow 1340
Scénarios associés 1342
tHiveClose **1343**
tHiveConnection **1344**
tHiveRow **1345**
tHSQLDbInput **1346**
Propriétés du tHSQLDbInput 1346
Scénarios associés 1348
tHSQLDbOutput **1349**
Propriétés du tHSQLDbOutput 1349
Scénarios associés 1352
tHSQLDbRow **1353**

Propriétés du tHSQLDbRow 1353
Scénarios associés 1355
tInterbaseClose 1356
Propriétés du tInterbaseClose 1356
Scénario associé 1356
tInterbaseCommit 1357
Propriétés du tInterbaseCommit 1357
Scénario associé 1357
tInterbaseConnection 1358
Propriétés du tInterbaseConnection ... 1358
Scénarios associés 1359
tInterbaseInput 1360
Propriétés du tInterbaseInput 1360
Scénarios associés 1361
tInterbaseOutput 1363
Propriétés du tInterbaseOutput 1363
Scénarios associés 1365
tInterbaseRollback 1367
Propriétés du tInterbaseRollback 1367
Scénario associé 1367
tInterbaseRow 1368
Propriétés du tInterbaseRow 1368
Scénarios associés 1370
tJavaDBInput 1371
Propriétés du tJavaDBInput 1371
Scénarios associés 1372
tJavaDBOutput 1373
Propriétés du tJavaDBOutput 1373
Scénarios associés 1376
tJavaDBRow 1377
Propriétés du tJavaDBRow 1377
Scénarios associés 1379
tJDBCColumnList 1380
Propriétés du tJDBCColumnList 1380
Scénario associé 1381
tJDBCClose 1382
Propriétés du tJDBCClose 1382
Scénario associé 1382
tJDBCCommit 1383
Propriétés du tJDBCCommit 1383
Scénario associé 1383
tJDBCConnection 1384
Propriétés du tJDBCConnection 1384
Scénario associé 1385
tJDBCInput 1386
Propriétés du tJDBCInput 1386
Le composant tJDBCInput dans les Jobs Talend Map/Reduce 1388
Scénarios associés 1389
tJDBCOutput 1390
Propriétés du tJDBCOutput 1390
Le composant tJDBCOutput dans les Jobs Talend Map/Reduce 1393
Le tJDBCOutput dans des Jobs Storm Talend 1394
Scénarios associés 1395
tJDBCRollback 1396
Propriétés du tJDBCRollback 1396
Scénario associé 1396
tJDBCRow 1397
Propriétés du tJDBCRow 1397
Scénarios associés 1399
tJDBCSP 1400
Propriétés du tJDBCSP 1400
Scénarios associés 1402
tJDBCTableList 1403
Propriétés du tJDBCTableList 1403
Scénario associé 1404
tLDAPAttributesInput 1405
Propriétés du tLDAPAttributesInput .. 1405
Scénario associé 1407
tLDAPClose 1408
Propriétés du tLDAPClose 1408
Scénario associé 1408
tLDAPConnection 1409
Propriétés du tLDAPConnection 1409
Scénario associé 1410
tLDAPInput 1411
Propriétés du tLDAPInput 1411
Scénario : Afficher le contenu filtré d'un annuaire LDAP 1413
tLDAPOutput 1416
Propriétés du tLDAPOutput 1416
Scénario : Editer des données dans un annuaire LDAP 1418
tLDAPRenameEntry 1422
Propriétés du tLDAPRenameEntry 1422
Scénarios associés 1424
tMaxDBInput 1425
Propriétés du tMaxDBInput 1425
Scénarios associés 1426
tMaxDBOutput 1427
Propriétés du tMaxDBOutput 1427
Scénarios associés 1429
tMaxDBRow 1430
Propriétés du tMaxDBRow 1430
Scénarios associés 1432
tParseRecordSet 1433
Propriétés du tParseRecordSet 1433
Scénario associé 1434
tPostgresPlusBulkExec 1435
Propriétés du tPostgresPlusBulkExec . 1435
Scénario associé 1437
tPostgresPlusClose 1438
Propriétés du tPostgresPlusClose 1438
Scénario associé 1438
tPostgresPlusCommit 1439
Propriétés du tPostgresPlusCommit ... 1439
Scénario associé 1439
tPostgresPlusConnection 1440
Propriétés du tPostgresPlusConnection 1440
Scénario associé 1441
tPostgresPlusInput 1442
Propriétés du tPostgresPlusInput 1442
Scénarios associés 1444
tPostgresPlusOutput 1445
Propriétés du tPostgresPlusOutput 1445
Scénarios associés 1448
tPostgresPlusOutputBulk 1449
Propriétés du tPostgresPlusOutputBulk 1449
Scénarios associés 1450
tPostgresPlusOutputBulkExec 1451
Propriétés du tPostgresplusOutputBulkExec 1451
Scénarios associés 1453
tPostgresPlusRollback 1454
Propriétés du tPostgresPlusRollback .. 1454
Scénario associé 1454
tPostgresPlusRow 1455
Propriétés du tPostgresPlusRow 1455
Scénarios associés 1457
tPostgresPlusSCD 1458
tPostgresPlusSCDELT 1459
tSAPHanaClose 1460
Propriétés du tSAPHanaClose 1460
Scénario associé 1460
tSAPHanaCommit 1461
Propriétés du tSAPHanaCommit 1461
Scénario associé 1461
tSAPHanaConnection 1462
Propriétés du tSAPHanaConnection ... 1462
Scénario associé 1463

tSAPHanaInput 1464
Propriétés du tSAPHanaInput 1464
Scénario associé 1466
tSAPHanaOutput 1467
Propriétés du tSAPHanaOutput 1467
Scénario associé 1470
tSAPHanaRollback 1471
Propriétés du tSAPHanaRollback 1471
Scénario associé 1471
tSAPHanaRow 1472
Propriétés du tSAPHanaRow 1472
Scénario associé 1474
tSasInput 1475
Propriétés du tSasInput 1475
Scénarios associés 1476
tSasOutput 1478
Propriétés du tSasOutput 1478
Scénarios associés 1480
tSQLiteClose 1481
Propriétés du tSQLiteClose 1481
Scénario associé 1481
tSQLiteCommit 1482
Propriétés du tSQLiteCommit 1482
Scénario associé 1482
tSQLiteConnection 1483
Propriétés du tSQLiteConnection 1483
Scénarios associés 1484
tSQLiteInput 1485
Propriétés du tSQLiteInput 1485
Scénario : Filtrer des données SQlite .. 1486
tSQLiteOutput 1489
Propriétés du tSQLiteOutput 1489
Scénario associé 1491
tSQLiteRollback 1492
Propriétés du tSQLiteRollback 1492
Scénario associé 1492
tSQLiteRow 1493
Propriétés du tSQLiteRow 1493
Scénario : Mettre à jour des lignes SQLite 1495
Scénario associé 1497
Composants DotNET 1499
tDotNETInstantiate 1500
Propriétés du tDotNETInstantiate 1500
Scénario associé 1501
tDotNETRow 1502
Propriétés du tDotNETRow 1502
Scénario : Utiliser .NET dans Talend . 1503
Composants ELT 1509
tAccessConnection 1510
tAS400Connection 1511
tCombinedSQLAggregate 1512
Propriétés du tCombinedSQLAggregate 1512
Scénario : Filtrer et agréger des colonnes d'une table directement sur le SGBD 1513
tCombinedSQLFilter 1518
Propriétés du tCombinedSQLFilter 1518
Scénario associé 1519
tCombinedSQLInput 1520
Propriétés du tCombinedSQLInput 1520
Scénario associé 1521
tCombinedSQLOutput 1522
Propriétés du tCombinedSQLOutput .. 1522
Scénario associé 1523
tDB2Connection 1524
tELTGreenplumInput 1525
Propriétés du tELTGreenplumInput ... 1525
Scénarios associés 1526
tELTGreenplumMap 1527
Propriétés du tELTGreenplumMap 1527
Scénario: Mapper les données à l'aide d'une jointure implicite simple .. 1529
Scénario associé 1535
tELTGreenplumOutput 1536
Propriétés du tELTGreenplumOutput 1536
Scénarios associés 1537
tELTJDBCInput 1538
Propriétés du tELTJDBCInput 1538
Scénarios associés 1539
tELTJDBCMap 1540
Propriétés du tELTJDBCMap 1540
Scénarios associés 1542
tELTJDBCOutput 1543
Propriétés du tELTJDBCOutput 1543
Scénarios associés 1544
tELTMSSqlInput 1545
Propriétés du tELTMSSqlInput 1545
Scénarios associés 1546
tELTMSSqlMap 1547
Propriétés du tELTMSSqlMap 1547
Scénarios associés 1549
tELTMSSqlOutput 1550
Propriétés du tELTMSSqlOutput 1550
Scénarios associés 1551
tELTMysqlInput 1552
Propriétés du tELTMysqlInput 1552
Scénarios associés 1553
tELTMysqlMap 1554
Propriétés du tELTMysqlMap 1554
Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre 1557
Scénario 2 : ELT utilisant une table Alias 1560
Scénario associé 1564
tELTMysqlOutput 1565
Propriétés du tELTMysqlOutput 1565
Scénarios associés 1566
tELTNetezzaInput 1567
Propriétés du tELTNetezzaInput 1567
Scénarios associés 1568
tELTNetezzaMap 1569
Propriétés du tELTNetezzaMap 1569
Scénarios associés 1571
tELTNetezzaOutput 1572
Propriétés du tELTNetezzaOutput 1572
Scénarios associés 1573
tELTOracleInput 1574
Propriétés du tELTOracleInput 1574
Scénario associé 1575
tELTOracleMap 1576
Propriétés du tELTOracleMap 1576
Scénario : Mettre à jour les lignes d'une base Oracle 1578
Scénario associé 1581
tELTOracleOutput 1582
Propriétés du tELTOracleOutput 1582
Scénario : Utiliser la fonction MERGE d'Oracle pour mettre à jour et insérer simultanément des données 1584
tELTPostgresqlInput 1589
Propriétés du tELTPostgresqlInput 1589
Scénarios associés 1590
tELTPostgresqlMap 1591
Propriétés du tELTPostgresqlMap 1591
Scénarios associés 1593
tELTPostgresqlOutput 1594
Propriétés du tELTPostgresqlOutput .. 1594
Scénarios associés 1595
tELTSybaseInput 1596
Propriétés du tELTSybaseInput 1596

Scénarios associés 1597
tELTSybaseMap 1598
Propriétés du tELTSybaseMap 1598
Scénarios associés 1600
tELTSybaseOutput 1601
Propriétés du tELTSybaseOutput 1601
Scénarios associés 1602
tELTTeradataInput 1603
Propriétés de tELTTeradataInput 1603
Scénarios associés 1604
tELTTeradataMap 1605
Propriétés de tELTTeradataMap 1605
Scénario : Mapper des données à l'aide d'une sous-requête 1607
Scénarios associés 1615
tELTTeradataOutput 1616
Propriétés de tELTTeradataOutput 1616
Scénarios associés 1617
tFirebirdConnection 1618
tGreenplumConnection 1619
tHiveConnection 1620
tIngresConnection 1621
tInterbaseConnection 1622
tJDBCConnection 1623
tMSSqlConnection 1624
tMysqlConnection 1625
tNetezzaConnection 1626
tOracleConnection 1627
tParAccelConnection 1628
tPostgresPlusConnection 1629
tPostgresqlConnection 1630
tSAPHanaConnection 1631
tSQLiteConnection 1632
tSQLTemplate 1633
Propriétés du tSQLTemplate 1633
Scénario associé 1634
tSQLTemplateAggregate 1635
Propriétés du tSQLTemplateAggregate 1635
Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD 1637
tSQLTemplateCommit 1641
Propriétés du tSQLTemplateCommit .. 1641
Scénario associé 1642
tSQLTemplateFilterColumns 1643
Propriétés du tSQLTemplateFilterColumns 1643
Scénario associé 1644
tSQLTemplateFilterRows 1645
Propriétés du tSQLTemplateFilterRows 1645
Scénario associé 1646
tSQLTemplateMerge 1647
Propriétés du tSQLTemplateMerge ... 1647
Scénario : Effectuer des opérations de MERGE directement dans le SGBD 1649
tSQLTemplateRollback 1656
Propriétés du tSQLTemplateRollback 1656
Scénario associé 1657
tSybaseConnection 1658
tTeradataConnection 1659
tVectorWiseConnection 1660
Composants ESB 1661
tESBConsumer 1662
Propriétés du tESBConsumer 1662
Scénario 1 : Utiliser le composant tESBConsumer 1666
Scénario 2 : Utiliser le tEBSConsumer avec des en-têtes SOAP personnalisés 1670
tESBProviderFault 1677
Propriétés du tESBProviderFault 1677
Scénario : Effectuer une requête sur les nom d'aéroports à partir des codes pays 1678
tESBProviderRequest 1689
Propriétés du tESBProviderRequest ... 1689
Scénario : Envoi d'un message via un service sans attente de réponse 1690
tESBProviderResponse 1700
Propriétés du tESBProviderResponse 1700
Scénario : Retourner une réponse "Hello world" 1701
tRESTClient 1710
Propriétés du tRESTClient 1710
Scénario 1 : Obtenir des informations sur un utilisateur en interagissant avec un service RESTful 1714
Scénario 2 : Mettre à jour les informations des utilisateurs via une interaction avec un service RESTful .. 1720
tRESTRequest 1727
Propriétés du tRESTRequest 1727
Scénario 1 : Service REST acceptant des requêtes HTTP GET et envoyant des réponses 1729
Scénario 2 : Utiliser les paramètres URI Query pour explorer les données d'une base de données 1734
Scénario 3 : Service REST acceptant des requêtes HTTP POST 1743
tRESTResponse 1750
Propriétés du tRESTResponse 1750
Scénario associé 1751
tRouteFault 1752
Propriétés du tRouteFault 1752
Scénario : Récupérer des messages du tRouteFault 1753
tRouteInput 1758
Propriétés du tRouteInput 1758
Scénario : Récupérer des messages d'une Route 1759
tRouteOutput 1763
Propriétés du tRouteOutput 1763
Scénario : Récupérer des messages du tRouteOutput 1764
Composants File 1769
tAdvancedFileOutputXML 1770
tApacheLogInput 1771
Propriétés du tApacheLogInput 1771
Scénario : Lire un fichier de log d'accès Apache 1772
tCreateTemporaryFile 1774
Propriétés du tCreateTemporaryFile ... 1774
Scénario : Créer un fichier temporaire et y écrire des données 1775
tChangeFileEncoding 1779
Propriétés du tChangeFileEncoding ... 1779
Scénario: Transformer l'encodage des caractères d'un fichier 1779
tFileArchive 1781
Propriétés du tFileArchive 1781
Scénario : Zipper des fichiers à l'aide d'un tFileArchive 1782
tFileCompare 1784
Propriétés du tFileCompare 1784
Scénario : Comparer des fichiers dézippés 1785

tFileCopy **1787**
Propriétés du tFileCopy 1787
Scénario : Récupérer un fichier de la corbeille 1788
tFileDelete **1790**
Propriétés du tFileDelete 1790
Scénario : Supprimer des fichiers 1791
tFileExist **1793**
Propriétés du tFileExist 1793
Scénario : Vérifier l'existence d'un fichier et le créer s'il n'existe pas 1794
tFileInputARFF **1798**
Propriétés du tFileInputARFF 1798
Scénario : Afficher le contenu d'un fichier ARFF 1799
tFileInputDelimited **1803**
Propriétés du tFileInputDelimited 1803
tFileInputDelimited in Talend Map/Reduce Jobs 1805
Scénario 1 : Afficher le contenu d'un fichier délimité 1807
Scénario 2 : Lire les données d'un fichier distant en mode stream 1809
tFileInputEBCDIC **1813**
Propriétés du tFileInputEBCDIC 1813
Scénario : Extraire les données d'un fichier EBCDIC 1814
tFileInputExcel **1821**
Propriétés du tFileInputExcel 1821
Scénario associé 1824
tFileInputFullRow **1825**
Propriétés du tFileInputFullRow 1825
Le tFileInputFullRow dans des Jobs Map/Reduce Talend 1826
Scénario : Lire des lignes complètes dans un fichier délimité 1828
tFileInputJSON **1830**
Propriétés du tFileInputJSON 1830
Le tFileInputJSON dans des Jobs Talend Map/Reduce 1832
Scénario 1 : Extraire des données JSON d'un fichier 1834
Scénario 2 : Extraire des données JSON d'un fichier en utilisant XPath .. 1837
Scénario 3 : Extraire des données JSON à partir d'une URL 1840
tFileInputLDIF **1845**
Propriétés du tFileInputLDIF 1845
Scénario associé 1846
tFileInputMail **1847**
Propriétés du tFileInputMail 1847
Scénario : Extraire des champs clés d'un e-mail 1848
tFileInputMSDelimited **1851**
Propriétés du tFileInputMSDelimited 1851
Scénario : Lecture d'un fichier délimité multi-structure 1853
tFileInputMSPositional **1860**
Propriétés du tFileInputMSPositional 1860
Scénario : Lire des données d'un fichier positionnel 1861
tFileInputMSXML **1865**
Propriétés du tFileInputMSXML 1865
Scénario : Lecture d'un fichier XML multi-structuré 1866
tFileInputPositional **1869**
Propriétés du tFileInputPositional 1869
Le tFileInputPositional dans des Jobs Map/Reduce Talend 1871
Scénario : Transformer un fichier positionnel en fichier XML 1873
tFileInputProperties **1878**
Propriétés du tFileInputProperties 1878
Scénario : Lire et mapper la clé et les valeurs de fichiers properties et alimenter un glossaire 1879
tFileInputRaw **1883**
Propriétés du tFileInputRaw 1883
Scénario associé 1884
tFileInputRegex **1885**
Propriétés du tFileInputRegex 1885
Scénario : Transformer en fichier Regex en Positional 1887
tFileInputXML **1890**
tFileList **1891**
Propriétés du tFileList 1891
Scénario 1 : Itération sur un répertoire 1893
Scénario 2 : Trouver des fichiers dupliqués entre deux dossiers 1895
tFileOutputARFF **1901**
Propriétés du tFileOutputARFF 1901
Scénario associé 1903
tFileOutputDelimited **1904**
Propriétés du tFileOutputDelimited ... 1904
Le tFileOutputDelimited dans des Jobs Map/Reduce Talend 1906
Scénario 1 : Ecrire des données dans un fichier délimité 1909
Scénario 2 : Utiliser un flux de sortie pour sauvegarder des données dans un fichier local 1913
tFileOutputEBCDIC **1916**
Propriétés du tFileOutputEBCDIC 1916
Scénario : Créer un fichier EBCDIC à partir de deux fichiers délimités 1917
tFileOutputExcel **1919**
Propriétés du tFileOutputExcel 1919
Scénario associé 1921
tFileOutputJSON **1922**
Propriétés du tFileOutputJSON 1922
Le tFileOutputJSON dans des Jobs Map/Reduce Talend 1923
Scénario : Ecrire un fichier JSON structuré 1925
tFileOutputLDIF **1928**
Propriétés du tFileOutputLDIF 1928
Scénario : Ecrire des données dans un fichier LDIF 1929
tFileOutputMSDelimited **1933**
Propriétés du tFileOutputMSDelimited 1933
Scénario associé 1934
tFileOutputMSPositional **1935**
Propriétés du tFileOutputMSPositional 1935
Scénario associé 1936
tFileOutputMSXML **1937**
Propriétés du tFileOutputMSXML 1937
Scénario associé 1942
tFileOutputPositional **1943**
Propriétés du tFileOutputPositional ... 1943
Le tFileOutputPositional dans des Jobs Map/Reduce Talend 1945
Scénario associé 1947
tFileOutputProperties **1948**
Propriétés du tFileOutputProperties ... 1948
Scénario associé 1949
tFileOutputRaw **1950**
Propriétés du tFileOutputRaw 1950
Scénario associé 1951

tFileOutputXML **1952**
tFileProperties **1953**
Propriétés du tFileProperties 1953
Scénario : Afficher les propriétés d'un fichier traité 1954
tFileRowCount **1956**
Propriétés du tFileRowCount 1956
Scénario : Ecrire un fichier dans MySQL si le nombre d'enregistrements correspond à une valeur de référence 1957
tFileTouch **1962**
Propriétés du tFileTouch 1962
Scénario associé 1962
tFileUnarchive **1963**
Propriétés du tFileUnarchive 1963
Scénario associé 1964
tGPGDecrypt **1965**
Propriétés du tGPGDecrypt 1965
Scénario : Décrypter un fichier crypté GnuPG et afficher son contenu 1965
tNamedPipeClose **1969**
Propriétés du tNamedPipeClose 1969
Scénario associé 1969
tNamedPipeOpen **1970**
Propriétés du tNamedPipeOpen 1970
Scénario associé 1970
tNamedPipeOutput **1971**
Propriétés du tNamedPipeOutput 1971
Scénario : Ecrire et charger des données via un tube nommé 1972
tPivotToColumnsDelimited **1978**
Propriétés du tPivotToColumnsDelimited 1978
Scénario : Utiliser une colonne pivot pour agréger des données 1979
Composants Internet **1983**
tFileFetch **1984**
Propriétés du tFileFetch 1984
Scénario 1 : Récupérer des données à partir d'un protocole HTTP 1986
Scénario 2 : Réutiliser un cookie stocké pour récupérer des fichiers via un protocole HTTP 1989
Scénario associé 1992
tFileInputJSON **1993**
tFTPConnection **1994**
Propriétés du tFTPConnection 1994
Scénarios associés 1995
tFTPDelete **1996**
Propriétés du tFTPDelete 1996
Scénarios associés 1997
tFTPFileExist **1998**
Propriétés du tFTPFileExist 1998
Scénarios associés 1999
tFTPFileList **2000**
Propriétés du tFTPFileList 2000
Scénario : Itération sur un répertoire distant 2001
tFTPFileProperties **2006**
Propriétés du tFTPFileProperties 2006
Scénario associé 2008
tFTPGet **2009**
Propriétés du tFTPGet 2009
Scénario associé 2011
tFTPPut **2012**
Propriétés du tFTPPut 2012
Scénario : Copier des fichiers sur un serveur FTP distant 2014
tFTPRename **2017**
Propriétés du tFTPRename 2017
Scénario : Renommer un fichier situé sur un serveur FTP 2018
Scénario associé 2020
tFTPTruncate **2021**
Propriétés du tFTPTruncate 2021
Scénario associé 2022
tHttpRequest **2023**
Propriétés du tHttpRequest 2023
Scénario 1 : Envoyer une requête HTTP au serveur et sauvegarder localement les réponses HTTP obtenues 2025
Scénario 2 : Envoyer une requête POST depuis un fichier local JSON ... 2026
tJMSInput **2031**
Propriétés du tJMSInput 2031
Scénario associé 2032
tJMSOutput **2033**
Propriétés du tJMSOutput 2033
Scénario : Mettre un message dans une file d'attente du serveur ActiveMQ et le retirer de cette file 2034
Scénario associé 2037
tMicrosoftMQInput **2038**
Propriétés du tMicrosoftMQInput 2038
Scénario : Ecrire et récupérer des messages dans une file d'attente de message Microsoft 2039
tMicrosoftMQOutput **2043**
Propriétés du tMicrosoftMQOutput ... 2043
Scénario associé 2044
tMomCommit **2045**
Propriétés du tMomCommit 2045
Scénario associé 2045
tMomConnection **2046**
Propriétés du tMomConnection 2046
Scénario associé 2047
tMomInput **2048**
Propriétés du tMomInput 2048
Scénario 1 : Communication asynchrone via un serveur MOM 2051
Scénario 2 : Transmettre des fichiers XML via un serveur MOM 2054
tMomMessageIdList **2060**
Propriétés du tMomMessageIdList 2060
Scénario associé 2061
tMomOutput **2062**
Propriétés du tMomOutput 2062
Scénario associé 2065
tMomRollback **2066**
Propriétés du tMomRollback 2066
Scénario associé 2066
tPOP **2067**
Propriétés du tPOP 2067
Scénario : Récupérer une sélection d'e-mails à partir d'un compte de messagerie 2069
tREST **2071**
Propriétés du tREST 2071
Scénario : Créer et récupérer des données en invoquant un service Web REST 2072
tRSSInput **2075**
Propriétés du tRSSInput 2075
Scénario : Récupérer les billets fréquemment mis à jour sur un blog ... 2076
tRSSOutput **2078**
Propriétés du tRSSOutput 2078
Scénario 1 : Créer un flux RSS et stocker les fichiers sur un serveur FTP 2079

Scénario 2 : Créer un flux RSS contenant des métadonnées 2083
Scénario 3 : Créer un fichier XML de flux ATOM 2085
tSCPClose 2089
Propriétés du tSCPClose 2089
Scénario associé 2089
tSCPConnection 2090
Propriétés du tSCPConnection 2090
Scénarios associés 2090
tSCPDelete 2091
Propriétés du tSCPDelete 2091
Scénario associé 2091
tSCPFileExists 2092
Propriétés du tSCPFileExists 2092
Gérer un fichier en utilisant SCP 2092
tSCPFileList 2098
Propriétés du tSCPFileList 2098
Scénario associé 2099
tSCPGet 2100
Propriétés du tSCPGet 2100
Scénario associé 2100
tSCPPut 2101
Propriétés du tSCPPut 2101
Scénario associé 2101
tSCPRename 2102
Propriétés du tSCPRename 2102
Scénario associé 2102
tSCPTruncate 2103
Propriétés du tSCPTruncate 2103
Scénario associé 2103
tSendMail 2104
Propriétés du tSendMail 2104
Scénario : Envoyer un e-mail lors d'une erreur 2105
tSetKerberosConfiguration 2109
Propriétés du tSetKerberosConfiguration 2109
Scénario associé 2109
tSetKeystore 2110
Propriétés du tSetKeystore 2110
Scénario : Extraire des informations client d'un fichier WSDL privé 2111
tSetProxy 2115
Propriétés du tSetProxy 2115
Scénario associé 2115
tSocketInput 2116
Propriétés du tSocketInput 2116
Scénario : Transférer des données vers un port d'écoute 2117
tSocketOutput 2120
Propriétés du tSocketOutput 2120
Scénario associé 2121
tSOAP 2122
Propriétés du tSOAP 2122
Scénario : Récupérer le nom d'un pays en utilisant un Service Web 2124
Scénario 2 : Utiliser un message SOAP depuis un fichier XML pour obtenir le nom d'un pays et le sauvegarder dans un fichier XML 2125
tSVNLogInput 2129
Propriétés du tSVNLogInput 2129
Scénario : Récupérer un message de log depuis un référentiel SVN 2130
tWebService 2132
Propriétés du tWebService 2132
Scénario : Obtenir des noms de pays à l'aide d'un service Web 2134
tWebServiceInput 2140
Propriétés du tWebServiceInput 2140
Scénario : Obtenir des noms de pays à l'aide d'un Webservice 2142
tXMLRPCInput 2144
Propriétés du tXMLRPCInput 2144
Scénario : Chercher le nom d'un Etat via une méthode XMLRPC 2145
Composants Logs & Errors 2147
tAssert 2148
Propriétés du tAssert 2148
Scénario 1: Obtenir le statut des commandes d'un produit (quotidiennement) par rapport à un nombre fixe 2148
Scénario 2 : Paramétrer une condition assertive pour l'exécution d'un Job 2152
tAssertCatcher 2158
Propriétés du tAssertCatcher 2158
Scénario associé 2159
tChronometerStart 2160
Propriétés du tChronometerStart 2160
Scénario associé 2160
tChronometerStop 2161
Propriétés du tChronometerStop 2161
Scénario : Mesurer la durée de traitement d'un sous-Job ou d'une partie d'un sous-Job 2162
tDie 2166
Propriétés du tDie 2166
Scénarios associés 2166
tFlowMeter 2168
Propriétés du tFlowMeter 2168
Scénario associé 2168
tFlowMeterCatcher 2169
Propriétés du tFlowMeterCatcher 2169
Scénario : Mesurer le flux d'un Job 2170
tLogCatcher 2174
Propriétés du tLogCatcher 2174
Scénario 1 : Avertissement & log lors de la création d'entrées 2174
Scénario 2 : Monitorer & arrêter un Job 2176
tLogRow 2178
Propriétés du tLogRow 2178
Scénarios associés 2180
tStatCatcher 2181
Propriétés du tStatCatcher 2181
Scénario : Afficher les statistiques d'exécution du Job 2182
tWarn 2185
Propriétés du tWarn 2185
Scénarios associés 2185
Composants Misc 2187
tAddLocationFromIP 2188
Propriétés du tAddLocationFromIP 2188
Scénario : Identifier la localisation géographique d'une adresse IP 2189
tBufferInput 2191
Propriétés du tBufferInput 2191
Scénario : Récupérer les données mises en mémoire tampon 2192
tBufferOutput 2194
Propriétés du tBufferOutput 2194
Scénario 1 : Mettre des données en mémoire tampon 2195
Scénario 2 : Mettre les données de sortie en mémoire tampon du serveur d'application Web 2197
tContextDump 2205
Propriétés du tContextDump 2205
Scénario associé 2205
tContextLoad 2206

Propriétés du tContextLoad 2206
Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement ... 2207
tFixedFlowInput 2214
Propriétés du tFixedFlowInput 2214
Scénarios associés 2215
tMemorizeRows 2216
Propriétés du tMemorizeRows 2216
Scénario : Compter le nombre d'occurrences de différents âges 2217
tMsgBox 2222
Propriétés du tMsgBox 2222
Scénario : Test de type 'Hello world!' 2222
tRowGenerator 2224
Propriétés du tRowGenerator 2224
Scénario : Génération aléatoire de fichiers de test 2226
tServerAlive 2229
Propriétés du tServerAlive 2229
Scénario : Valider le statut de la connexion à un hôte distant 2229
Composants Orchestration 2233
tFileList 2234
tFlowToIterate 2235
Propriétés du tFlowToIterate 2235
Scénario : Transformer un flux de données en liste 2236
tForeach 2240
Propriétés du tForeach 2240
Scénario : Faire une boucle sur une liste et récupérer les valeurs 2240
tInfiniteLoop 2243
Propriétés du tInfiniteLoop 2243
Scénario associé 2243
tIterateToFlow 2244
Propriétés du tIterateToFlow 2244
Scénario : Transformer une liste de fichiers en flux de données 2245
tLoop 2247
Propriétés du tLoop 2247
Scénario : Exécuter un Job à partir d'une boucle 2248
tPostjob 2251
Propriétés du tPostjob 2251
Scénario associé 2251
tPrejob 2252
Propriétés du tPrejob 2252
Scénario : Gérer des fichiers avant et après l'exécution d'un Job 2252
Scénario associé 2255
tReplicate 2256
Propriétés du tReplicate 2256
Scénario : Répliquer un flux et trier deux flux identiques 2257
tRunJob 2262
tUnite 2263
Propriétés du tUnite 2263
Scénario : Itération sur des fichiers et fusion de contenu 2264
tSleep 2267
Propriétés du tSleep 2267
Scénario associé 2267
tWaitForFile 2268
Propriétés du tWaitForFile 2268
Scénario 1 : Attendre la création d'un fichier et arrêter l'itération après le déclenchement d'un message 2269
Scénario 2 : Attendre la création d'un fichier et poursuivre l'itération après le déclenchement d'un message 2271
tWaitForSocket 2273
Propriétés du tWaitForSocket 2273
Scénario associé 2273
tWaitForSqlData 2274
Propriété du tWaitForSqlData 2274
Scénario : Attendre l'ajout de lignes dans une table 2275
Composants Processing 2279
tAggregateRow 2280
Propriétés du tAggregateRow 2280
Scénario : Agréger des valeurs et trier des données 2282
tAggregateSortedRow 2285
Propriétés du tAggregateSortedRow .. 2285
Scénario associé 2286
tConvertType 2287
Propriétés du tConvertType 2287
Scénario 1 : Convertir des types de données Java 2288
Scénario 2 : Convertir des types Java à l'aide de composants Map/Reduce ... 2292
tDenormalize 2301
Propriétés du tDenormalize 2301
Scénario 1 : Dénormaliser une colonne 2302
Scénario 2 : Dénormaliser plusieurs colonnes 2303
tDenormalizeSortedRow 2306
Propriétés du tDenormalizeSortedRow 2306
Scénario : Regrouper des lignes triées 2307
tExternalSortRow 2310
Propriétés du tExternalSortRow 2310
Scénario associé 2311
tExtractDelimitedFields 2312
Propriétés du tExtractDelimitedFields 2312
Le tExtractDelimitedFields dans des Jobs Map/Reduce Talend 2313
Scénario : Extraire des champs à partir d'un fichier délimité par des virgules 2315
tExtractEBCDICFields 2319
Propriétés du tExtractEBCDICFields . 2319
Scénario 2320
tExtractJSONFields 2321
Propriétés du tExtractJSONFields 2321
Scénario : Récupérer les messages d'erreur lors de l'extraction de données de champs JSON 2323
Scénario 2 : Collecter des données de votre réseau social favori 2329
tExtractPositionalFields 2333
Propriétés du tExtractPositionalFields 2333
Scénario associé 2335
tExtractRegexFields 2336
Propriétés du tExtractRegexFields 2336
Scénario : Extraire des noms, des domaines et domaines de premier niveau à partir d'adresses e-mail 2338
tExtractXMLField 2341
tFilterColumns 2342
Propriétés du tFilterColumns 2342
Scénario associé 2343
tFilterRow 2344
Propriétés du tFilterRow 2344

Scénario 1 : Filtrer une liste de noms à l'aide de conditions simples 2346
Scénario 2 : Filtrer une liste de noms via différentes opérations logiques 2349
tJoin 2351
Propriétés du tJoin 2351
Scénario 1 : Faire une correspondance exacte entre deux colonnes et écrire les données rejetées 2352
tMap 2357
Propriétés du tMap 2357
Scénario 1 : Mapping simple de données avec filtre et jointure explicite 2358
Scénario 2 : Mapping de données avec rejet Inner join 2362
Scénario 3 : Mapping de jointure en cascade 2367
Scénario 4 : Mapping avancé avec filtres, jointures explicites et rejet Inner join 2367
Scénario 5 : Mapping avancé avec filtres et différents rejets 2372
Scénario 6 : Mapping avancé avec rechargement de la base de référence à chaque ligne 2375
Scénario 7 : Mapping avec des tables de sortie 2382
tNormalize 2387
Propriétés du tNormalize 2387
Scénario 1 : Normaliser des données .. 2388
Scénario 2 : Normaliser des données à l'aide de composants Map/Reduce ... 2391
tReplace 2401
Propriétés du tReplace 2401
Scénario : Remplacement multiple et filtrage de colonnes 2402
Scénario 2 : Remplacer des valeurs et filtrer des colonnes à l'aide de composants Map/Reduce 2405
tSampleRow 2415
Propriétés du tSampleRow 2415
Scénario : Filtrer des lignes et des groupes de lignes 2416
tSortRow 2419
Propriétés du tSortRow 2419
Scénario : Trier des données 2420
tSplitRow 2423
Propriétés du tSplitRow 2423
Scénario 1 : Séparer une ligne en plusieurs lignes 2424
tWriteJSONField 2428
Propriétés du tWriteJSONField 2428
Scénario : Ecrire des données plates dans des champs JSON 2430
Scénarios associés 2434
tXMLMap 2435
Propriétés du tXMLMap 2435
Scénario 1 : Mapper et transformer des données de source XML 2436
Scénario 2 : Créer un flux Lookup pour effectuer une jointure sur des données complémentaires XML 2441
Scénario 3 : Mapper des données à l'aide d'un filtre 2446
Scénario 4 : Capturer les données rejetées par le flux Lookup et le filtre 2449
Scénario 5 : Mapper des données à l'aide d'un élément "group" 2452
Scénario 6 : Classer les données de sortie avec l'élément "aggregate" 2456
Scénario 7 : Restructurer des données produit à l'aide de différents éléments de boucle 2459
Composants System 2469
tRunJob 2470
Propriétés du tRunJob 2470
Scénario 1 : Exécuter un Job fils 2472
Scénario 2 : Exécuter une liste de Jobs fils de manière dynamique 2477
Scénario 3 : Propager les données de sortie en mémoire tampon du Job fils au Job père 2480
tSetEnv 2485
Propriétés du tSetEnv 2485
Scénario : Modifier une variable lors de l'exécution d'un Job 2485
tSSH 2489
Propriétés du tSSH 2489
Scénario : Accéder aux informations du système distant via SSH 2491
tSystem 2493
Propriétés du tSystem 2493
Scénario : Echo ‘Hello World!' 2495
Composants Talend MDM 2497
tMDMBulkLoad 2498
Propriétés du tMDMBulkLoad 2498
Scénario : Charger des enregistrements dans une entité métier 2501
tMDMClose 2507
Propriétés du tMDMClose 2507
Scénario associé 2507
tMDMCommit 2508
Propriétés du tMDMCommit 2508
Scénario associé 2508
tMDMConnection 2509
Propriétés du tMDMConnection 2509
Scénario associé 2510
tMDMDelete 2511
Propriétés du tMDMDelete 2511
Scénario : Supprimer des données maître d'un hub MDM 2513
tMDMInput 2520
Propriétés du tMDMInput 2520
Scénario : Lire les données maître d'un hub MDM 2522
tMDMOutput 2526
Propriétés du tMDMOutput 2526
Scénario : Ecrire des données maître dans un hub MDM 2529
tMDMReceive 2535
Propriétés du tMDMReceive 2535
Scénario : Extraire des informations d'un enregistrement MDM au format XML 2536
tMDMRollback 2541
Propriétés du tMDMRollback 2541
Scénario associé 2541
tMDMRouteRecord 2542
Propriétés du tMDMRouteRecord 2542
Scénario : Router un enregistrement vers l'Event Manager 2543
tMDMSP 2551
Propriétés du tMDMSP 2551
Scénario : Exécuter une procédure stockée dans le hub MDM 2552
tMDMTriggerInput 2557
Propriétés du tMDMTriggerInput 2557

Scénario : Échanger des informations concernant les événements d'un enregistrement MDM 2558
tMDMTriggerOutput 2570
Propriétés du tMDMTriggerOutput ... 2570
Scénario associé 2571
tMDMViewSearch 2572
Propriétés du tMDMViewSearch 2572
Scénario : Récupérer des enregistrements d'un hub MDM via une vue existante 2574
Composants Technical 2579
tHashInput 2580
Propriétés du tHashInput 2580
Scénario 1 : Lire des données directement dans la mémoire cache afin d'y accéder rapidement 2581
Scénario 2 : Vider la mémoire avant d'y charger les données si une boucle existe dans le même sous-job 2584
tHashOutput 2589
Propriétés du tHashOutput 2589
Scénarios associés 2590
Composants XML 2591
tAdvancedFileOutputXML 2592
Propriétés du tAdvancedFileOutputXML 2592
Scénario : Créer un fichier XML à l'aide d'une boucle 2599
tDTDValidator 2604
Propriétés du tDTDValidator 2604
Scénario : Valider un fichier XML 2604
tEDIFACTtoXML 2607
Propriétés du tEDIFACTtoXML 2607
Scénario : Passer d'EDIFACT à XML 2607
tExtractXMLField 2610
Propriétés du tExtractXMLField 2610
Scénario 1 : Extraire les données XML d'un champ d'une table de base de données 2611
Scénario 2 : Extraire les données valides et les données erronées à partir d'un champ XML dans fichier délimité 2614
tFileInputXML 2618
Propriétés du tFileInputXML 2618
Le tFileInputXML dans des Jobs Map/Reduce Talend 2620
Scénario 1 : Extraire des adresses XML 2623
Scénario 2 : Extraire les données XML erronées dans un flux de rejet ... 2624
tFileOutputXML 2628
Propriétés de tFileOutputXML 2628
Le tFileOutputXML dans des Jobs Map/Reduce Talend 2630
Scénarios associés 2632
tWriteXMLField 2633
Propriétés du tWriteXMLField 2633
Scénario : Extraire la structure d'un fichier XML et l'insérer dans les champs d'une base de données 2635
tXMLMap 2639
tXSDValidator 2640
Propriétés du tXSDValidator 2640
Scénario : Valider des flux de données par rapport à un fichier XSD 2641
tXSLT 2644
Propriétés du tXSLT 2644
Scénario : Transformer un fichier XML en HTML à l'aide d'une feuille de style XSL 2645

Préface

Informations générales

Objectif

Ce Guide de référence propose des scénarios et détaille comment paramétrer les principaux composants accessibles à partir de chacun des différents groupes de la **Palette** dans la perspective **Integration** du *Studio Talend*.

Le Guide de référence des composants **Talend** est la version française non exhaustive du *Components Reference Guide* anglais. Pour les composants non documentés, référez-vous à ce dernier. Veillez à consulter régulièrement le site Internet de **Talend** pour télécharger la dernière version du *Guide de référence* français.

Les informations contenues dans ce document s'appliquent à la version **5.6.3** du *Studio Talend*.

Public visé

Ce guide s'adresse aux utilisateurs et aux administrateurs du *Studio Talend*.

Dans ce document, l'interface présentée par le biais des captures d'écran peut être légèrement différente de la vôtre.

Conventions typographiques

Ce guide utilise les conventions typographiques suivantes :

- texte en **gras** : boutons et champs d'une fenêtre ou d'une boîte de dialogue, touches du clavier, menus et options des menus ;
- texte en **[gras]** : titre des fenêtres, assistants et boîtes de dialogue ;
- texte en `courier` : paramètres système renseignés par l'utilisateur ;
- texte en *italique* : nom des fichiers, schémas, colonnes, lignes ou variables dont il est question dans les exemples utilisés, ainsi que le nom des champs des vues Basic settings et Advanced settings dont il est fait mention dans le tableau des propriétés de chaque composant.
- L'icône introduit un paragraphe apportant une information supplémentaire sur un point important. Elle introduit également les commentaires relatifs à un tableau ou une image.
- L'icône introduit un message apportant une information relative aux modalités d'exécution, ou des recommandations particulières. Elle est également utilisée pour attirer l'attention de l'utilisateur sur certaines informations ou situations particulièrement importantes.

Remarques et Support

Votre opinion nous intéresse, n'hésitez pas à nous faire part de vos remarques, suggestions, requêtes concernant la documentation ou le produit, et obtenez le support de notre équipe **Talend**, sur le Forum **Talend** (en anglais) disponible à l'adresse :

http://talendforge.org/forum

Composants Big Data

Ce chapitre passe en revue les principaux composants présents dans la famille **Big Data** de la **Palette**.

L'utilisation de ces composants vous permet de vous connecter aux modules de la distribution d'Hadoop que vous utilisez dans un environnement de développement unifié et d'effectuer des opérations nativement sur des clusters Big Data.

Les composants **Big Data** permettent de créer des connexions à différents outils tiers utilisés pour le transfert, le stockage ou l'analyse de Big Data, tels que Sqoop, MongoDB ou BigQuery. Ils vous permettent de charger, extraire, transformer et traiter rapidement des ensembles de données volumineux et variés.

Votre opinion nous intéresse, n'hésitez pas à nous faire part de vos remarques, suggestions, requêtes concernant la documentation ou le produit, et obtenez le support de notre équipe **Talend**, sur le Forum **Talend** (en anglais) disponible à l'adresse : http://www.talendforge.org/forum/index.php

tGSBucketCreate

Propriétés du tGSBucketCreate

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSBucketCreate** crée un nouveau bucket dans Google Cloud Storage.	
Objectif	Ce composant permet de créer un nouveau bucket que vous pouvez utiliser afin d'organiser des données et contrôler les accès aux données dans Google Cloud Storage.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key et *Secret Key*	Saisissez les informations d'authentification obtenues de Google afin d'effectuer des requêtes sur Google Cloud Storage. Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour plus d'informations sur la clé d'accès et la clé secrète, rendez-vous sur https://developers.google.com/storage/docs/reference/v1/getting-startedv1?hl=en/ (en anglais) et consultez la description des clés développeur. *Les champs **Access Key** et **Secret Key** ne sont disponibles que si la case **Use an existing connection** n'est pas cochée.*
Bucket configuration	*Bucket name*	Spécifiez le nom du bucket que vous souhaitez créer. Notez que le nom du bucket doit être unique au sein du système Google Cloud Storage. Pour plus d'informations sur les conventions de nommage des buckets, consultez https://developers.google.com/storage/docs/bucketnaming (en anglais).
	Special configure	Cochez cette case afin de fournir une configuration supplémentaire au bucket à créer.
	Project ID	Spécifiez l'identifiant du projet auquel le nouveau bucket appartient.
	Location	Sélectionnez dans la liste l'emplacement du nouveau bucket. Actuellement, **Europe** et **US** sont disponibles. Par défaut, l'emplacement du bucket est **US**. Notez qu'une fois le bucket créé à un emplacement spécifique, il ne peut être déplacé vers un nouvel emplacement.
	Acl	Dans la liste, sélectionnez la liste de contrôle d'accès (ACL) souhaitée pour le nouveau bucket. Selon l'ACL du bucket, les accès requis par les utilisateurs sont acceptés ou rejetés. Si vous ne spécifiez pas d'ACL prédéfinie pour le nouveau bucket, l'ACL privée prédéfinie du projet s'applique. Pour plus d'informations sur les ACL, consultez https://developers.google.com/storage/docs/accesscontrol?hl=en (en anglais).
	Die on error	Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec un **tGSBucketList** afin de vérifier si le nouveau bucket est correctement créé.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	n/a	

Scénario associé

Pour des scénarios associés, consultez *Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3* .

tGSBucketDelete

Propriétés du tGSBucketDelete

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSBucketDelete** supprime un bucket dans Google Cloud Storage.	
Objectif	Ce composant supprime un bucket vide dans Google Cloud Storage afin de libérer des ressources occupées.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key et *Secret Key*	Saisissez les informations d'authentification obtenues de Google afin d'effectuer des requêtes sur Google Cloud Storage. Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour plus d'informations sur la clé d'accès et la clé secrète, rendez-vous sur https://developers.google.com/storage/docs/reference/v1/getting-startedv1?hl=en/ (en anglais) et consultez la description des clés développeur. *Les champs **Access Key** et **Secret Key** ne sont disponibles que si la case **Use an existing connection** n'est pas cochée.*
	Bucket name	Spécifiez le nom du bucket que vous souhaitez supprimer. Assurez-vous que le bucket à supprimer est vide. *La suppression du bucket ne peut être annulée. Vous devez donc vous assurer d'avoir sauvegardé toutes les données que vous souhaitez garder avant de continuer.*
	Die on error	Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec un **tGSBucketList** afin de vérifier si les données sont bien importées.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	n/a

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tGSBucketExist

Propriétés du tGSBucketExist

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSBucketExist** vérifie que le bucket spécifié existe dans Google Cloud Storage.	
Objectif	Ce composant permet de vérifier l'existence d'un bucket afin d'effectuer plus d'opérations.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key et *Secret Key*	Saisissez les informations d'authentification obtenues de Google afin d'effectuer des requêtes sur Google Cloud Storage. Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour plus d'informations sur la clé d'accès et la clé secrète, rendez-vous sur https://developers.google.com/storage/docs/reference/v1/getting-startedv1?hl=en/ (en anglais) et consultez la description des clés développeur. *Les champs **Access Key** et **Secret Key** ne sont disponibles que si la case **Use an existing connection** n'est pas cochée.*
	Bucket name	Spécifiez le nom du bucket dont vous souhaitez vérifier l'existence dans Google Cloud Storage.
	Die on error	Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**BUCKET_EXIST** : présence ou non d'un bucket spécifié. Cette variable est une variable **Flow** et retourne un booléen. **BUCKET_NAME** : nom d'un bucket spécifié. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisable en tant que composant standalone.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	n/a

Scénario associé

Pour des scénarios associés, *Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3* .

tGSBucketList

Propriétés du tGSBucketList

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSBucketList** effectue une boucle sur tous les buckets de tous les projets ou d'un projet spécifique dans Google Cloud Storage.	
Objectif	Ce composant vous permet de récupérer une liste des buckets de tous les projets ou d'un projet spécifique dans Google Cloud Storage.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key et *Secret Key*	Saisissez les informations d'authentification obtenues de Google afin d'effectuer des requêtes sur Google Cloud Storage. Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour plus d'informations sur la clé d'accès et la clé secrète, rendez-vous sur https://developers.google.com/storage/docs/reference/v1/getting-startedv1?hl=en/ (en anglais) et consultez la description des clés développeur. *Les champs **Access Key** et **Secret Key** ne sont disponibles que si la case **Use an existing connection** n'est pas cochée.*
	Specify project ID	Cochez cette case et, dans le champ **Project ID**, spécifiez l'identifiant du projet dont vous voulez récupérer la liste des buckets.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**CURRENT_BUCKET_NAME** : nom du bucket courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_BUCKET** : nombre de buckets. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisable en tant que composant standalone ou en tant que composant de début d'un Job.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	n/a

Scénario associé

Pour un scénario associé, consultez *Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3* .

tGSClose

Propriétés du tGSClose

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSClose** ferme une connexion active sur Google Cloud Storage.	
Objectif	Ce composant permet de fermer une connexion à Google Cloud Storage afin de libérer les ressources occupées.	
Basic settings	*Component List*	Dans la liste, sélectionnez le composant **tGSConnection** si plus d'une connexion est prévue pour le Job courant.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Google Cloud Storage, en particulier le **tGSConnection**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	n/a	

Scénario associé

Pour un scénario dans lequel le **tGSClose** est utilisé, consultez *Scénario : Gérer des fichiers avec Google Cloud Storage*.

tGSConnection

Propriétés du tGSConnection

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSConnection** ouvre une connexion à Google Cloud Storage.	
Objectif	Ce composant permet de fournir les informations d'authentification nécessaires pour effectuer des requêtes sur le système Google Cloud Storage et permet de réutiliser la connexion créée pour Google Cloud Storage.	
Basic settings	*Access Key* et *Secret Key*	Saisissez les informations d'authentification obtenues de Google afin d'effectuer des requêtes sur Google Cloud Storage. Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour plus d'informations sur la clé d'accès et la clé secrète, rendez-vous sur https://developers.google.com/storage/docs/reference/v1/getting-startedv1?hl=en/ (en anglais) et consultez la description des clés développeur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Google Cloud Storage, en particulier le **tGSClose**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	n/a	

Scénario associé

Pour un scénario dans lequel le **tGSConnection** est utilisé, consultez *Scénario : Gérer des fichiers avec Google Cloud Storage*.

tGSCopy

Propriétés du tGSCopy

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSCopy** copie ou déplace des objets au sein d'un bucket ou entre plusieurs buckets dans Google Cloud Storage.	
Objectif	Ce composant permet de rationaliser les traitements en effectuant des tâches de copie de manière automatique.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key et *Secret Key*	Saisissez les informations d'authentification obtenues de Google afin d'effectuer des requêtes sur Google Cloud Storage. Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour plus d'informations sur la clé d'accès et la clé secrète, rendez-vous sur https://developers.google.com/storage/docs/reference/v1/getting-startedv1?hl=en/ (en anglais) et consultez la description des clés développeur. *Les champs **Access Key** et **Secret Key** ne sont disponibles que si la case **Use an existing connection** n'est pas cochée.*
	Source bucket name	Saisissez le nom du bucket à partir duquel vous souhaitez copier ou déplacer des objets.
	Source object key	Saisissez la clé de l'objet à copier.
	Source is folder	Cochez cette case si l'objet source est un dossier.
	Target bucket name	Saisissez le nom du bucket vers lequel vous souhaitez copier ou déplacer des objets.
	Target folder	Saisissez le dossier cible vers lequel les objets sont copiés ou déplacés.
	Action	Sélectionnez l'action que vous souhaitez effectuer sur les objets de la liste. • **Copy** : copie les objets depuis le bucket ou le dossier source vers le bucket ou dossier cible. • **Move** : déplace les objets depuis le bucket ou dossier source vers le bucket ou dossier cible.
	Rename	Cochez cette case et, dans le champ **New name**, saisissez un nouveau nom pour l'objet à copier ou à déplacer. *La case **Rename** n'est pas disponible si vous cochez la case **Source is folder**.*
	Die on error	Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.

Global Variables	**SOURCE_BUCKET** : nom du bucket source. Cette variable est une variable **After** et retourne une chaîne de caractères. **SOURCE_OBJECTKEY** : clé d'un objet source. Cette variable est une variable **After** et retourne une chaîne de caractères. **DESTINATION_BUCKETNAME** : nom du bucket cible. Cette variable est une variable **After** et retourne une chaîne de caractères. **DESTINATION_FOLDER** : dossier cible. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	n/a

Scénario associé

Pour un scénario dans lequel le **tGSCopy** est utilisé, consultez *Scénario : Gérer des fichiers avec Google Cloud Storage*.

tGSDelete

Propriétés du tGSDelete

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSDelete** supprime les objets qui correspondent aux critères définis dans Google Cloud Storage.	
Objectif	Ce composant permet de supprimer des objets depuis Google Cloud Storage afin de libérer des ressources occupées.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key et *Secret Key*	Saisissez les informations d'authentification obtenues de Google afin d'effectuer des requêtes sur Google Cloud Storage. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google. Pour plus d'informations sur la clé d'accès et la clé secrète, rendez-vous sur https://developers.google.com/storage/docs/reference/v1/getting-startedv1?hl=en/ (en anglais) et consultez la description des clés développeur. *Les champs **Access Key** et **Secret Key** ne sont disponibles que si la case **Use an existing connection** n'est pas cochée.*
	Key prefix	Saisissez le préfixe de la clé des objets à supprimer. Ainsi, seuls les objets dont la clé commence par ce préfixe sont listés.
	Delimiter	Saisissez le séparateur afin de supprimer uniquement les objets avec des noms jusqu'à ce séparateur.
	Specify project ID	Cochez cette case et, dans le champ **Project ID**, saisissez l'identification du projet dont vous voulez supprimer des objets.
	Delete object from bucket list	Cochez cette case et renseignez la table **Bucket** pour supprimer des objets dans les buckets spécifiés. • **Bucket name** : saisissez le nom du bucket dans lequel vous souhaitez supprimer les objets. • **Key prefix** : saisissez le préfixe pour supprimer les objets dont les clés commencent par la chaîne définie dans le bucket spécifié. • **Delimiter** : saisissez le séparateur afin de lister uniquement les objets avec des noms jusqu'à ce séparateur. *Si vous cochez la case **Delete object from bucket list**, les champs **Key prefix** et **Delimiter**, ainsi que la case **Specify project ID** ne sont pas disponibles.*
	Die on error	Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.

Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé avec le composant **tGSList** afin de vérifier si les objets correspondant aux critères définis sont correctement supprimés.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	n/a

Scénario associé

Pour un scénario dans lequel le **tGSDelete** est utilisé, consultez *Scénario : Gérer des fichiers avec Google Cloud Storage*.

tGSGet

tGSGet properties

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSGet** récupère les objets qui correspondent aux critères définis dans Google Cloud Storage et les envoie vers un dossier local.	
Objectif	Ce composant permet de télécharger des fichiers depuis Google Cloud Storage vers un dossier local.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant adéquat dans la liste **Component List** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key et *Secret Key*	Saisissez les informations d'authentification obtenues de Google afin d'effectuer des requêtes sur Google Cloud Storage. Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour plus d'informations sur la clé d'accès et la clé secrète, rendez-vous sur https://developers.google.com/storage/docs/reference/v1/getting-startedv1?hl=en/ (en anglais) et consultez la description des clés développeur. *Les champs **Access Key** et **Secret Key** ne sont disponibles que si la case **Use an existing connection** n'est pas cochée.*
	Key prefix	Saisissez le préfixe afin de télécharger uniquement les objets dont les clés commencent par le préfixe défini.
	Delimiter	Saisissez le séparateur afin de télécharger uniquement les objets avec des noms jusqu'à ce séparateur.
	Specify project ID	Cochez cette case et, dans le champ **Project ID**, saisissez l'identification du projet dont vous voulez obtenir des objets.
	Use keys	Cochez cette case et renseignez la table **Keys** afin de définir les critères pour les objets à télécharger depuis Google Cloud Storage. • **Bucket name** : saisissez le nom du bucket à partir duquel vous souhaitez télécharger des objets. • **Key** : saisissez la clé de l'objet à télécharger. • **New name** : saisissez un nouveau nom pour l'objet à télécharger. *Si vous cochez la case **Use keys**, les champs **Key prefix** et **Delimiter**, ainsi que les cases **Specify project ID** et **Get files from bucket list** ne sont pas disponibles.*
	Get files from bucket list	Cochez cette case et renseignez la table **Bucket** afin de définir les critères pour les objets à télécharger depuis Google Cloud Storage. • **Bucket name** : saisissez le nom du bucket à partir duquel vous souhaitez télécharger des objets. • **Key prefix** : saisissez le préfixe pour télécharger des objets dont les clés commencent par le préfixe spécifié dans le bucket défini.

		• **Delimiter** : saisissez le séparateur afin de télécharger les objets avec des noms jusqu'à ce séparateur depuis le bucket défini. *Si vous cochez la case **Get files from bucket list**, les champs **Key prefix** et **Delimiter**, ainsi que les cases **Specify project ID** et **Use keys** ne sont pas disponibles.*
	Output directory	Spécifiez le dossier dans lequel vous souhaitez stocker les objets téléchargés.
	Die on error	Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Google Cloud Storage, par exemple le **tGSPut**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	n/a	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tGSList

Propriétés du tGSList

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSList** fait une boucle sur une liste d'objets qui correspondent aux critères spécifiés dans Google Cloud Storage.	
Objectif	Ce composant permet de récupérer, un à un, une liste d'objets depuis Google Cloud Storage.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant adéquat dans la liste **Component List** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key et *Secret Key*	Saisissez les informations d'authentification obtenues de Google afin d'effectuer des requêtes sur Google Cloud Storage. Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour plus d'informations sur la clé d'accès et la clé secrète, rendez-vous sur https://developers.google.com/storage/docs/reference/v1/getting-startedv1?hl=en/ (en anglais) et consultez la description des clés développeur. *Les champs **Access Key** et **Secret Key** ne sont disponibles que si la case **Use an existing connection** n'est pas cochée.*
	Key prefix	Saisissez le préfixe de la clé des objets à lister. Ainsi, seuls les objets dont la clé commence par ce préfixe sont listés.
	Delimiter	Saisissez le séparateur afin de lister uniquement les objets avec des noms jusqu'à ce séparateur.
	Specify project ID	Cochez cette case et, dans le champ **Project ID**, saisissez l'identification du projet dont vous voulez récupérer la liste des objets.
	List objects in bucket list	Cochez cette case et renseignez la table **Bucket** pour récupérer les objets dans les buckets spécifiés. • **Bucket name** : saisissez le nom du bucket à partir duquel vous souhaitez récupérer les objets. • **Key prefix** : saisissez le préfixe pour lister uniquement les objets dont les clés commencent par la chaîne définie dans le bucket spécifié. • **Delimiter** : saisissez le séparateur afin de lister uniquement les objets avec des noms jusqu'à ce séparateur. *Si vous cochez la case **List objects in bucket list**, les champs **Key prefix** et **Delimiter**, ainsi que la case **Specify project ID** ne sont pas disponibles .*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**CURRENT_BUCKET** : nom du bucket courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_KEY** : clé courante. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Le composant **tGSList** peut être utilisé en standalone ou en tant que composant de début.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	n/a

Scénario associé

Pour un scénario dans lequel le **tGSList** est utilisé, consultez *Scénario : Gérer des fichiers avec Google Cloud Storage*.

tGSPut

Propriétés du tGSPut

Famille de composant	Big Data / Google Cloud Storage	
Fonction	Le composant **tGSPut** met un fichier d'un dossier local dans Google Cloud Storage.	
Objectif	Ce composant est conçu pour charger des fichiers dans Google Cloud Storage afin que vous puissiez les gérer avec Google Cloud Storage.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key et *Secret Key*	Saisissez les informations d'authentification obtenues de Google pour envoyer des requêtes à Google Cloud Storage. Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour plus d'informations sur la clé d'accès et la clé secrète, rendez-vous sur https://developers.google.com/storage/docs/reference/v1/getting-startedv1?hl=en/ (en anglais) et consultez la description des clés développeur. *Les champs **Access Key** et **Secret Key** ne sont disponibles que si la case **Use an existing connection** n'est pas cochée.*
	Bucket name	Saisissez le nom du bucket dans lequel vous souhaitez charger les fichiers.
	Local directory	Saisissez le chemin d'accès complet ou parcourez votre système vers le dossier local contenant les fichiers à charger.
	Google Storage directory	Saisissez le nom du dossier Google Storage dans lequel vous souhaitez charger des fichiers.
	Use files list	Cochez cette case et renseignez la table **Files**. • **Filemask** : saisissez un nom ou un masque de fichier à l'aide de caractères spéciaux (*) ou d'expressions régulières. • **New name**: saisissez le nom à donner au fichier après le chargement.
	Die on error	Cette case est décochée par défaut afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé avec d'autres composants Google Cloud Storage, par exemple le **tGSGet**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	n/a

Scénario : Gérer des fichiers avec Google Cloud Storage

Le scénario suivant permet de créer un Job chargeant des fichiers à partir d'un répertoire local dans un bucket dans Google Storage. Ce Job permet ensuite d'effectuer des opérations de copie, de déplacement et de suppression sur ces fichiers, puis de lister et afficher les fichiers dans leurs buckets respectifs dans la console.

Prérequis : Vous devez posséder un compte Google Cloud Storage et avoir créé trois buckets dans le même répertoire Google Cloud Storage. Dans cet exemple, les buckets créés sont **bighouse**, **bed_room** et **study_room**.

Déposer et relier les composants

Afin de créer ce Job, procédez comme suit :

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tGSConnection**, un **tGSPut**, deux **tGSCopy**, un **tGSDelete**, un **tGSList**, un **tIterateToFlow**, un **tLogRow** et un **tGSClose**.
2. Reliez le **tGSConnection** au **tGSPut** à l'aide d'un lien **Trigger > On Subjob Ok**.
3. Reliez le **tGSPut** au premier **tGSCopy** à l'aide d'un lien **Trigger > On Subjob Ok**.
4. Reliez ensuite le premier **tGSCopy** au second **tGSCopy**, le **tGSCopy** au **tGSDelete**, le **tGSDelete** au **tGSList** et le **tGSList** au **tGSClose** à l'aide de liens **Trigger > On Subjob Ok**.
5. Reliez le **tGSList** au **tIterateToFlow** à l'aide d'un lien **Row > Iterate**.
6. Reliez le **tIterateToFlow** au **tLogRow** à l'aide d'un lien **Row > Main**.

Configurer les composants

Ouvrir une connexion à Google Cloud Storage

1. Double-cliquez sur le **tGSConnection** afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Naviguez vers la console d'API de Google dans votre navigateur afin d'accéder au projet Google contenant les services Cloud Storage que vous souhaitez utiliser.
3. Cliquez sur Google Cloud Storage > Interoperable Access afin d'ouvrir la vue correspondante et copiez la clé d'accès et la clé secrète dans votre presse-papier.
4. Dans la vue **Component** du Studio, collez la clé d'accès et la clé secrète dans les champs correspondants, respectivement **Access Key** et **Secret Key**.

Charger les fichiers dans Google Cloud Storage

1. Double-cliquez sur le **tGSPut** afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Cochez la case **Use an existing connection** puis sélectionnez la connexion que vous avez configurée précédemment.

3. Dans le champ **Bucket name**, saisissez le nom du bucket dans lequel vous souhaitez charger les fichiers, *bighouse* dans cet exemple.

4. Dans le champ **Local directory**, parcourez votre système vers le dossier contenant les fichiers à charger, *D:/Input/House* dans cet exemple.

   La capture d'écran suivante montre les fichiers contenus dans ce dossier :

5. Laissez les autres paramètres tels qu'ils sont.

Copier tous les fichiers d'un bucket à un autre

1. Double-cliquez sur le premier **tGSCopy** afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Cochez la case **Use an existing connection** puis sélectionnez la connexion que vous avez configurée précédemment.

3. Dans le champ **Source bucket name**, saisissez le nom du bucket à partir duquel vous souhaitez copier les fichiers, *bighouse* dans cet exemple.

4. Cochez la case **Source is a folder**. Tous les fichiers contenus dans le bucket *bighouse* sont copiés.

5. Dans le champ **Target bucket name**, saisissez le nom du bucket vers lequel vous souhaitez copier les fichiers, *bed_room* dans cet exemple.

6. Sélectionnez **Copy** dans la liste **Action**.

Déplacer un fichier d'un bucket à un autre et les renommer

1. Double-cliquez sur le second **tGSCopy** afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Cochez la case **Use an existing connection** puis sélectionnez la connexion que vous avez configurée précédemment.

3. Dans le champ **Source bucket name**, saisissez le nom du bucket contenant le fichier que vous souhaitez déplacer, *bighouse* dans cet exemple.

4. Dans le champ **Source object key**, saisissez la clé de l'objet à déplacer, *computer_01.txt* dans cet exemple.

5. Dans le champ **Target bucket name**, saisissez le nom du bucket vers lequel vous souhaitez déplacer le fichier, *study_room* dans cet exemple.

6. Sélectionnez **Move** dans la liste **Action**. Le fichier source défini, *computer_01.txt*, est déplacé du bucket *bighouse* vers le bucket *study_room*.

7. Cochez la case **Rename**. Dans le champ **New name**, saisissez un nouveau nom pour le fichier déplacé, *laptop.txt* dans cet exemple.

8. Laissez les autres paramètres tels qu'ils sont.

Supprimer un fichier dans un bucket

1. Double-cliquez sur le **tGSDelete**, afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Cochez la case **Use an existing connection** puis sélectionnez la connexion que vous avez configurée précédemment.

3. Cochez la case **Delete object from bucket list**. Dans la table **Bucket**, saisissez les informations du fichier que vous souhaitez supprimer.

   Dans cet exemple, le fichier *computer_03.csv* est supprimé du bucket *bed_room* dont les fichiers sont copiés depuis le bucket *bighouse*.

Lister les fichiers contenus dans les trois buckets

1. Double-cliquez sur le **tGSList** afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Cochez la case **Use an existing connection** puis sélectionnez la connexion que vous avez configurée précédemment.

3. Cochez la case **List objects in bucket list**. Dans la colonne *Bucket name* de la table **Bucket**, saisissez le nom des trois buckets, *bighouse*, *study_room* et *bed_room*.

4. Double-cliquez sur le **tIterateToFlow** afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

5. Cliquez sur le bouton **Edit schema** afin de définir les données qui sont passées au **tLogRow**.

   Dans cet exemple, ajoutez deux colonnes, *bucketName* et *key*, de type **Object**.

6. La table **Mapping** est automatiquement renseignée avec les colonnes définies.

   Dans la colonne *Value*, saisissez *globalMap.get("tGSList_2_CURRENT_BUCKET")* dans la ligne *bucketName* et *globalMap.get("tGSList_2_CURRENT_KEY")* dans la ligne *key*, respectivement. Vous pouvez également appuyer sur **Ctrl + Espace** puis choisir les variables appropriées.

7. Double-cliquez sur le **tLogRow** afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

8. Sélectionnez l'option **Table (print values in cells of a table)** pour un meilleur affichage des résultats.

Fermer la connexion à Google Cloud Storage

1. Double-cliquez sur le **tGSClose** afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Dans la liste **Component List**, sélectionnez la connexion que vous souhaitez fermer.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder le Job.

2. Appuyez sur **F6** ou cliquez sur **Run** dans l'onglet **Run** afin d'exécuter le Job.

```
.----------+---------------.
|        tLogRow_1         |
|=---------+--------------=|
|bucketName|key             |
|=---------+--------------=|
|bighouse  |books.txt       |
|bighouse  |clothes.txt     |
|bighouse  |computer_02.txt|
|bighouse  |computer_03.csv|
|bighouse  |model cars.txt  |
|bighouse  |tables.txt      |
|bighouse  |toys.txt        |
|study_room|laptop.txt      |
|bed_room  |books.txt       |
|bed_room  |clothes.txt     |
|bed_room  |computer_01.txt|
|bed_room  |computer_02.txt|
|bed_room  |model cars.txt  |
|bed_room  |tables.txt      |
|bed_room  |toys.txt        |
'----------+---------------'
```

Les fichiers contenus dans les trois buckets s'affichent. Comme attendu, les fichiers du bucket *bighouse* sont d'abord copiés vers le bucket *bed_room*. Ensuite, le fichier *computer_01.txt* du bucket *bighouse* est déplacé vers le bucket *study_room* puis renommé *laptop.txt*. Enfin, le fichier *computer_03.csv* est supprimé du bucket *bed_room*.

tHiveClose

Propriétés du tHiveClose

Famille de composant	Big Data/Hive	
Fonction	Le composant **tHiveClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Hive.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tHiveConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Hive, notamment le **tHiveConnection**, car le **tHiveConnection** vous permet d'ouvrir une connexion pour la transaction à effectuer.	
Prérequis	La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d'exemple relatives à MapR. • Assurez-vous d'avoir installé le client MapR sur la même machine que le *Studio Talend* et d'avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D'après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées *MAPR_INSTALL\ hadoop\hadoop-VERSION\lib\native*. Par exemple, pour Windows, la bibliothèque est *lib\MapRClient.dll* dans le fichier Jar du client MapR. Pour plus d'informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n'ajoutez pas de librairie, il est possible que vous rencontriez l'erreur suivante : `no MapRClient in java.library.path`. • Configurez l'argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]**. Cet argument fournit	

	au studio le chemin d'accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d'utiliser entièrement l'aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR. Pour plus d'informations concernant la configuration de cet argument, consultez la section expliquant comment visualiser les données dans le *Guide de prise en main de Talend Open Studio for Big Data*. Pour plus d'informations concernant l'installation d'une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tHiveConnection

Propriétés du tHiveConnection

Famille de composant	Big Data/Hive	
Fonction	Le composant **tHiveConnection** ouvre une connexion vers une base de données Hive.	
Objectif	Ce composant vous permet d'établir une connexion Hive à réutiliser dans d'autres composants Hive de votre Job.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
Version	*Distribution*	Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique. • Si disponible dans la liste de **Distribution**, l'option **Microsoft HD Insight** vous permet d'utiliser un cluster Microsoft HD Insight. Dans cette optique, vous devez configurez les connexions aux services WebHCat, HD Insight et Windows Azure Storage du cluster dans les zones affichées. Une vidéo démontrant comment configurer cette connexion est disponible à l'adresse suivante : https://www.youtube.com/watch?v=A3QTT6VsNoM. • L'option **Custom** vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**. Pour vous connecter à une distribution personnalisée, une fois l'option **Custom** sélectionnée, cliquez sur le bouton ... pour afficher une fenêtre, dans laquelle vous pouvez : 1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou, 2. Sélectionner **Import from zip** pour importer un fichier .zip contenant une distribution personnalisée que vous pouvez télécharger depuis http://www.talendforge.org/exchange/index.php. Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d'importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant. Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez *Connexion à une distribution Hadoop personnalisée*.
	Hive version	Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez. En plus de l'évolution de Hadoop, veuillez noter les changements suivants :

		• Si vous utilisez **Hortonworks Data Platform V2.2**, les fichiers de configuration de votre cluster peuvent utiliser des variables d'environnement comme *${hdp.version}*. Si vous êtes dans cette situation, vous devez configurer la propriété *mapreduce.application.framework.path* dans la table **Hadoop properties** de ce composant, avec la valeur du chemin d'accès pointant explicitement vers l'archive du framework MapReduce de votre cluster. Par exemple : `mapreduce.application.framework.path=/hdp/apps/2.2.0.0-2041/mapreduce/mapreduce.tar.gz#mr-framework` • Si vous utilisez **Hortonworks Data Platform V2.0.0**, le type de système d'exploitation utilisé pour la distribution Hadoop et pour le Job **Talend** doit être identique, par exemple Windows ou Linux. Sinon, vous devez utiliser le Jobserver de **Talend** afin d'exécuter un Job dans le même type de système d'exploitation que celui dans lequel la distribution **Hortonworks Data Platform V2.0.0** utilisée est exécutée. Pour plus d'informations concernant le Jobserver de **Talend**, consultez le *Guide d'installation et de migration Talend*.
	Connection mode	Sélectionnez un mode de connexion dans la liste. Les options varient selon la distribution que vous utilisez.
	Hive server	Sélectionnez le serveur Hive sur lequel vous souhaitez que le Job utilisant ce composant exécute des requêtes dans Hive. La liste **Hive server** est disponible uniquement lorsque la distribution Hadoop à utiliser, par exemple **HortonWorks Data Platform V1.2.0 (Bimota)** supporte HiveServer2. Vous pouvez sélectionner HiveServer2 (**Hive 2**), le serveur supportant mieux les connexions simultanées de différents clients que HiveServer (**Hive 1**). Pour plus d'informations concernant HiveServer2, consultez la page https://cwiki.apache.org/confluence/display/Hive/Setting+Up+HiveServer2 (en anglais).
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données. Ce champ n'est pas disponible lorsque vous sélectionnez **Embedded** dans la liste **Connection mode**.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Authentication	*Use kerberos authentication*	Si vous accédez au Metastore de Hive avec une sécurité Kerberos, cochez cette case et saisissez ensuite les paramètres appropriés dans les champs qui s'affichent. Les valeurs des paramètres suivants peuvent être trouvées dans le fichier *hive-site.xml* du système Hive utilisé. 1. **Hive principal** utilise la valeur de *hive.metastore.kerberos.principal*. C'est le principal du service du Metastore de Hive. 2. **Metastore URL** utilise la valeur de *javax.jdo.option.ConnectionURL*. C'est la chaîne JDBC de connexion au Metastore de Hive. 3. **Driver class** utilise la valeur de *javax.jdo.option.ConnectionDriverName*. C'est le nom du pilote de la connexion JDBC.

		4. **Username** utilise la valeur de *javax.jdo.option.ConnectionUserName*. Ce paramètre, ainsi que le paramètre **Password**, sont utilisés pour les informations de connexion de l'utilisateur au Metastore de Hive. 5. **Password** utilise la valeur de *javax.jdo.option.ConnectionPassword*. Cette case est disponible ou indisponible selon la distribution d'Hadoop à laquelle vous vous connectez.
	Use a keytab to authenticate	Cochez la case **Use a keytab to authenticate** pour vous connecter à un système Hadoop utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**. Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.
	Use SSL encryption	Cochez cette case pour activer la connexion cryptée SSL. Les champs qui s'affichent ensuite fournissent les informations d'authentification : • Dans le champ **Trust store path**, saisissez le chemin ou parcourez votre système jusqu'au fichier TrustStore à utiliser. Par défaut, les types TrustStore supportés sont *JKS* et *PKCS 12*. • Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Cette fonctionnalité n'est disponible que pour HiveServer2 en mode **Standalone** pour les distributions suivantes : • Hortonworks Data Platform 2.0 + • Cloudera CDH4 + • Pivotal HD 2.0 +
Hadoop properties	*Set Jobtracker URI*	Cochez cette case pour indiquer l'emplacement du service JobTracker à utiliser, dans le cluster Hadoop. Par exemple, si vous avez choisi une machine nommée *machine1* comme JobTracker, définissez son emplacement comme suit *machine1:portnumber*. Notez qu'un JobTracker est le service qui assigne des tâches Map/Reduce à des nœuds spécifiques dans un cluster Hadoop. Le mot "Job" dans JobTracker ne désigne pas un Job **Talend** mais un job Hadoop, décrit comme job MR ou MapReduce dans la documentation de Apache Hadoop : http://hadoop.apache.org (en anglais). Cette propriété est requise lorsque la requête que vous souhaitez utiliser est exécutée dans Windows et est une requête `Select`. Par exemple, `SELECT your_column_name FROM your_table_name` Si vous utilisez YARN dans votre cluster Hadoop, comme **Hortonworks Data Platform V2.0.0** ou **Cloudera CDH4.3 + (mode YARN)**, vous devez spécifier l'emplacement du **Resource Manager** et non celui du Jobtracker. Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d'un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du runtime) :

		1. Cochez la case **Set resourcemanager scheduler address** et saisissez l'adresse de l'ordonnanceur (Scheduler) dans le champ qui apparaît. 2. Allouez des volumes de mémoire corrects aux calculs **Map** et **Reduce** ainsi qu'à l'application **ApplicationMaster** de YARN en cochant la case **Set memory** dans la vue **Advanced settings**. 3. Cochez la case **Set jobhistory address** et saisissez l'emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory. 4. Cochez la case **Set staging directory** et saisissez le chemin d'accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l'exécution de programmes. Ce répertoire se trouve sous la propriété *yarn.app.mapreduce.am.staging-dir* dans les fichiers de configuration, notamment les fichiers *yarn-site.xml* et *mapred-site.xml* de votre distribution. 5. Cochez la case **Set Hadoop user** et saisissez le nom de l'utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d'écriture, ce champ vous permet d'exécuter le Job directement avec l'utilisateur ayant les droits d'accès appropriés au fichier ou répertoire à traiter. 6. Cochez la case **Use datanode hostname** pour permettre au Job d'accéder aux nœuds de données via leurs noms d'hôte. Cela définit la propriété *dfs.client.use.datanode.hostname* en *true*. Pour plus d'informations concernant le framework Hadoop Map/Reduce, consultez le tutoriel Map/Reduce dans la documentation de Apache Hadoop : http://hadoop.apache.org (en anglais).
	Set NameNode URI	Cochez cette case pour indiquer l'emplacement du NameNode du cluster Hadoop à utiliser. Le NameNode est le nœud maître d'un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée *masternode* comme NameNode d'une distribution Apache Hadoop, son emplacement est *hdfs://masternode:portnumber*. Cette propriété est requise lorsque la requête que vous souhaitez utiliser est exécutée dans Windows et est une requête `Select`. Par exemple, `SELECT your_column_name FROM your_table_name` Pour plus d'informations concernant le framework Hadoop Map/Reduce, consultez le tutoriel Map/Reduce dans la documentation de Apache Hadoop : http://hadoop.apache.org (en anglais).
Microsoft HD Insight properties	*WebHCat configuration*	Saisissez l'adresse et les informations d'authentification du service WebHCat du cluster Microsoft HD Insight à utiliser. Le studio utilise ce service pour soumettre le Job au cluster HD Insight. Dans le champ **Job result folder**, saisissez l'emplacement où vous souhaitez stocker les résultats d'exécution du Job dans Azure Storage.
	HDInsight configuration	Saisissez les informations d'authentification information du cluster HD Insight à utiliser.
	Windows Azure Storage configuration	Saisissez l'adresse et les informations d'authentification du compte Azure Storage à utiliser. Dans le champ **Container**, saisissez le nom du conteneur à utiliser. Dans le champ **Deployment Blob**, saisissez l'emplacement où vous souhaitez stocker le Job et ses bibliothèques dépendantes dans le compte Azure Storage.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de

		données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
HBase Configuration	*Store by HBase*	Cochez cette case afin d'afficher les paramètres à configurer pour permettre aux composants Hive d'accéder aux tables HBase : • Une fois l'accès configuré, vous pourrez utiliser, dans un **tHiveRow** et un **tHiveInput**, les instructions Hive QL permettant de lire et d'écrire des données dans HBase. • Si vous utilisez l'authentification Kerberos, vous devez définir les principaux relatifs à HBase dans les champs correspondants qui sont affichés. Pour plus d'informations à propos de cet accès concernant Hive et HBase, consultez la documentation de Apache Hive concernant l'intégration Hive/HBase.
	Zookeeper quorum	Saisissez le nom ou l'URL du service Zookeeper utilisé pour coordonner les transactions entre **Talend** et HBase. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer la propriété **zookeeper.znode.parent** pour définir la racine du chemin relatif d'un fichier HBase Zookeeper puis de sélectionner la case **Set Zookeeper znode parent** afin de définir cette propriété.
	Zookeeper client port	Saisissez le numéro du port d'écoute client du service Zookeeper que vous utilisez.
	Define the jars to register for HBase	Cochez cette case pour afficher la table **Register jar for HBase**, dans laquelle vous pouvez enregistrer tout fichier Jar manquant, requis pour HBase, par exemple, Hive Storage Handler, enregistré par défaut avec votre installation Hive.
	Register jar for HBase	Cliquez sur le bouton pour ajouter des lignes à la table, puis, dans la colonne **Jar name**, sélectionnez le(s) fichier(s) Jar à enregistrer. Dans la colonne **Jar path**, saisissez le chemin d'accès à ce(s) Jar(s).
Advanced settings	*Hadoop properties*	Le *Studio Talend* utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l'exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut. • Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**. Pour plus d'informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation Apache Hadoop sur le site http://hadoop.apache.org (en anglais) ou la documentation de la distribution Hadoop que vous utilisez. A titre d'exemple, les liens vers certaines propriétés sont listés ci-après: • Généralement, les propriétés relatives à HDFS peuvent être trouvées dans le fichier *hdfs-default.xml*

		correspondant à votre distribution, comme par exemple http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais). • Apache fournit également une page listant toutes les propriétés relatives à Hive : http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais).
	Hive properties	Le *Studio Talend* utilise la configuration par défaut pour son moteur afin d'effectuer des opération dans un base de données Hive. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez cette table avec la (les) propriété(s) à personnaliser. Ensuite, à l'exécution, la (les) propriété(s) personnalisée(s) écrasent celles par défaut. Pour plus d'informations concernant les propriétés dédiées à Hive, consultez https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration (en anglais). • Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.
	Mapred job map memory mb et *Mapred job reduce memory mb*	Si le système Hadoop utilisé est Hortonworks Data Platform V1.2 ou Hortonworks Data Platform V1.3, vous devez définir des allocations de mémoire suffisantes pour que les opérations de map et reduce soient effectuées par le système Hadoop. Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations *map* et *reduce* dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux *1000*, ce qui est normalement adapté pour l'exécution de ces opérations.
	Path separator in server	Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d'autres termes, changez le séparateur si celui-ci n'est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Hive, notamment le composant **tHiveClose**. Si le *Studio Talend* utilisé pour vous connecter à une base de données Hive fonctionne sous Windows, vous devez créer manuellement un dossier appelé *tmp* à la racine du disque où le *Studio Talend* est installé.	
Prérequis	La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d'exemple relatives à MapR. • Assurez-vous d'avoir installé le client MapR sur la même machine que le *Studio Talend* et d'avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D'après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées *MAPR_INSTALL\ hadoop\hadoop-VERSION\lib\native*. Par exemple, pour Windows, la bibliothèque est *lib\MapRClient.dll* dans le fichier Jar du client MapR. Pour	

	plus d'informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n'ajoutez pas de librairie, il est possible que vous rencontriez l'erreur suivante : `no MapRClient in java.library.path`. • Configurez l'argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]**. Cet argument fournit au studio le chemin d'accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d'utiliser entièrement l'aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR. Pour plus d'informations concernant la configuration de cet argument, consultez la section expliquant comment visualiser les données dans le *Guide de prise en main de Talend Open Studio for Big Data*. Pour plus d'informations concernant l'installation d'une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Connexion à une distribution Hadoop personnalisée

Comme expliqué dans le tableau des propriétés, lorsque vous sélectionnez l'option **Custom** dans la liste déroulante **Distribution**, vous pouvez vous connectez à une distribution qui n'est pas dans la liste **Distribution**.

Une fois l'option **Custom** sélectionnée, cliquez sur le bouton ... pour afficher la boîte de dialogue **[Import custom definition]**, puis procédez comme suit :

1. Selon votre cas, sélectionnez **Import from existing version** ou **Import from zip** afin de configurer la distribution Hadoop personnalisée à laquelle vous souhaitez vous connecter.

   - Si vous avez le fichier .zip de la distribution Hadoop à laquelle vous souhaitez vous connecter, sélectionnez **Import from zip**. La communauté **Talend** fournit ce type de fichiers .zip que vous pouvez télécharger depuis http://www.talendforge.org/exchange/index.php.

   - Sinon, sélectionnez **Import from existing version** pour importer une distribution de base officiellement supportée afin de la personnaliser à l'aide de l'assistant suivant.

Notez que les cases de cet assistant vous permettent de sélectionner le(s) élément(s) Hadoop que vous souhaitez importer. Selon le contexte dans lequel vous créez la connection, toutes les cases ne sont pas disponibles. Par exemple, si vous créez cette connexion pour un composant Hive, seule la case **Hive** est disponible.

2. Que vous ayez sélectionné **Import from existing version** ou **Import from zip**, vérifiez que chaque case à côté de l'élément Hadoop que vous souhaitez importer est cochée.

3. Cliquez sur **OK** et, dans la fenêtre d'avertissement, cliquez sur **Yes** pour accepter d'écraser toute configuration personnalisée des fichiers Jar précédemment implémentée.

   La boîte de dialogue **[Custom Hadoop version definition]** devient active.

Cette boîte de dialogue liste les éléments Hadoop et les fichiers Jar que vous importez.

4. Si vous avez coché l'option **Import from zip**, cliquez sur **OK** afin de valider la configuration importée.

   Si avez sélectionné l'option **Import from existing version**, afin d'importer une distribution de base, vous devez également importer des fichiers jar pour personnaliser cette distribution. Dans l'onglet de l'élément Hadoop que vous souhaitez personnaliser, par exemple l'onglet **HDFS/HCatalog/Oozie**, cliquez sur le bouton **[+]** pour ouvrir la boîte de dialogue **[Select libraries]**.

5. Sélectionnez l'option **External libraries** pour ouvrir sa vue.

6. Cliquez sur le bouton **Browse...** et parcourez votre système jusqu'au fichier Jar que vous souhaitez importer.

7. Cliquez sur **OK** pour valider les modifications et fermer la boîte de dialogue **[Select libraries]**.

   Le fichier Jar sélectionné apparaît dans la liste de l'onglet de l'élément Hadoop configuré.

   Notez que si vous souhaitez partager la configuration personnalisée Hadoop avec un autre *Studio Talend*, vous pouvez exporter cette connexion personnalisée de la boîte de dialogue **[Custom Hadoop version definition]** en cliquant sur le bouton .

8. Dans la boîte de dialogue **[Custom Hadoop version definition]**, cliquez sur **OK** afin de valider la configuration personnalisée et retourner à la liste **Distribution** de la vue **Basic settings** du composant.

Une fois la configuration de la distribution Hadoop personnalisée terminée vous pouvez, depuis la vue **Basic settings**, continuer de saisir d'autres paramètres requis pour la connexion.

Si la distribution Hadoop personnalisée à laquelle vous souhaitez vous connecter contient YARN, cochez la case **Use YARN** à côté de la liste **Distribution**.

Scénario : Créer une table Hive partitionnée

Ce scénario explique comment utiliser un composant **tHiveConnection**, un composant **tHiveCreateTable** et un **tHiveLoad** afin de créer une table Hive partitionnée et d'y écrire des données.

Notez que les composants **tHiveCreateTable** et **tHiveLoad** ne sont disponibles que si vous utilisez une des solutions **Talend** avec Big Data.

Les données d'exemple à utiliser dans ce scénario sont des informations relatives aux employés d'une entreprise et se présentent comme suit :

```
1;Lyndon;Fillmore;21-05-2008;US
2;Ronald;McKinley;15-08-2008
3;Ulysses;Roosevelt;05-10-2008
4;Harry;Harrison;23-11-2007
5;Lyndon;Garfield;19-07-2007
6;James;Quincy;15-07-2008
7;Chester;Jackson;26-02-2008
8;Dwight;McKinley;16-07-2008
9;Jimmy;Johnson;23-12-2007
10;Herbert;Fillmore;03-04-2008
```

Les informations contiennent le nom de certains employés et la date de leur enregistrement dans le système des ressources humaines. Comme ces employés travaillent pour la filiale américaine de l'entreprise, créez une partition *US* pour ces données.

Avant de commencer à reproduire ce scénario, assurez-vous d'avoir les droits d'accès appropriés à la base de données Hive à utiliser.

Notez que si vous utilisez le système d'exploitation Windows, vous devez créer un dossier *tmp* à la racine du disque où le studio est installé.

Procédez comme suit :

Relier les composants

1. Dans la perspective **Integration** du studio, créez un Job vide depuis le nœud **Job Designs** du **Repository**.

   Pour plus d'informations concernant la création de Jobs, consultez le chapitre relatif dans le *Guide utilisateur du Studio Talend*.

2. Déposez un **tHiveConnection**, un **tHiveCreateTable** et un **tHiveLoad** dans l'espace de modélisation graphique.

3. Reliez-les à l'aide de liens **Trigger** > **OnSubjobOk**.

Configurer la connexion à Hive

Configurer le tHiveConnection

1. Double-cliquez sur le **tHiveConnection** pour ouvrir sa vue **Component**.

2. Dans la liste **Property type**, sélectionnez **Built-in**. Si vous avez créé la connexion à utiliser dans le **Repository**, sélectionnez **Repository**, cliquez sur le bouton ... pour ouvrir la boîte de dialogue **[Repository content]** et sélectionnez cette connexion. Ainsi, le studio réutilise les paramètres de connexion dans le Job.

   Pour plus d'informations concernant la création d'une connexion Hadoop dans le **Repository**, consultez le chapitre décrivant le nœud **Hadoop cluster** dans le *Guide de prise en main de Talend Open Studio for Big Data*.

3. Dans la zone **Version**, sélectionnez la distribution de Hadoop à utiliser, ainsi que sa version. Si votre distribution n'est pas dans la liste, sélectionnez **Custom** pour vous connecter à une distribution de Hadoop non officiellement supportée dans le studio.

   Pour comprendre comment utiliser l'option **Custom** étape par étape, consultez *Connexion à une distribution Hadoop personnalisée*.

4. Dans la zone **Connection**, saisissez les paramètres de connexion à la base de données Hive à utiliser.

5. Dans le champ **Name node**, saisissez l'emplacement du nœud maître, le NameNode, de la distribution à utiliser. Par exemple, saisissez *talend-hdp-all:50300*.

6. Dans le champ **Job tracker**, saisissez l'emplacement du JobTracker de votre distribution. Par exemple, saisissez *hdfs://talend-hdp-all:8020*.

   La notion de Job dans le terme de JobTracker désigne les Jobs MR ou MapReduce décrits dans la documentation de Apache sur le site http://hadoop.apache.org/ (en anglais).

Créer la table Hive

Définir le schéma

1. Double-cliquez sur le **tHiveCreateTable** pour ouvrir sa vue **Component**.

2. Cochez la case **Use an existing connection** et, dans la liste **Component list**, sélectionnez la connexion configurée dans le composant **tHiveConnection** utilisé dans ce Job.

3. Cliquez sur le bouton ... à côté du champ **Edit schema** pour ouvrir l'éditeur de schéma.

4. Cliquez quatre fois sur le bouton pour ajouter quatre lignes dans la colonne **Column**. Renommez-les respectivement *Id*, *FirstName*, *LastName* et *Reg_date*.

Notez que vous ne pouvez pas utilisez les mots-clés réservés à Hive pour nommer les colonnes, comme par exemple *location* ou *date*.

5. Dans la colonne **Type**, sélectionnez le type de données dans chaque colonne. Dans ce scénario, *Id* est de type **Integer**, *Reg_date* est de type **Date** et les autres lignes sont de type **String**.
6. Dans la colonne **DB type**, sélectionnez le type Hive de chaque colonne correspondant au type de données défini. Par exemple, *Id* est de type **INT** et *Reg_date* est de type **TIMESTAMP**.
7. Dans la colonne **Data pattern**, définissez le modèle correspondant à celui des données brutes. Dans cet exemple, utilisez le modèle par défaut.
8. Cliquez sur **OK** pour valider ces modifications.

Configurer les paramètres de la table

1. Dans le champ **Table name**, saisissez le nom de la table Hive à créer. Dans ce scénario, saisissez *employees*.
2. Dans la liste **Action on table**, sélectionnez **Create table if not exists**.
3. Dans la liste **Format**, sélectionnez le format des données pour lequel la table Hive est créée. Dans ce scénario, sélectionnez **TEXTFILE**.
4. Cochez la case **Set partitions** pour ajouter la partition *US* comme expliqué au début de ce scénario. Pour définir cette partition, cliquez sur le bouton ... à côté du champ **Edit schema**.
5. Laissez la case **Set file location** décochée pour utiliser le chemin par défaut pour la table Hive.
6. Cochez la case **Set Delimited row format** pour afficher les options disponibles du format de ligne.
7. Cochez la case **Field** et saisissez un point-virgule (;) comme séparateur de champs dans le champ qui apparaît.
8. Cochez la case **Line** et laissez la valeur par défaut comme séparateur de lignes.

Ecrire des données dans la table

Configurer le tHiveLoad

1. Double-cliquez sur le **tHiveLoad** pour ouvrir sa vue **Component**.

2. Cochez la case **Use an existing connection** et, dans la liste **Component list**, sélectionnez la connexion configurée dans le composant **tHiveConnection** utilisé dans ce Job.

3. Dans le champ **Load action**, sélectionnez **LOAD** pour écrire des données du fichier d'exemple présenté au début du scénario.

4. Dans le champ **File path**, saisissez le chemin d'accès au répertoire dans lequel sont stockées les données. Dans cet exemple, les données sont stockées sur le système HDFS utilisé.

   Notez que si vous devez lire des données depuis un système de fichiers local autre que le système HDFS, vous devez vous assurer que les données lues sont stockées dans le système de fichiers local de la machine sur laquelle le Job est exécuté. Cochez ensuite la case **Local** dans la vue **Basic settings**. Par exemple, lorsque le mode de connexion à Hive est **Standalone**, le Job est exécuté sur la machine sur laquelle l'application Hive est installée. Les données sont donc stockées sur cette machine.

5. Dans le champ **Table name**, saisissez le nom de la table cible dans laquelle charger les données. Dans ce scénario, saisissez *employees*.

6. Dans la liste **Action on file**, sélectionnez **APPEND**.

7. Cochez la case **Set partitions** et, dans le champ qui apparaît, saisissez le nom de la partition dans laquelle vous souhaitez ajouter des données. Dans ce scénario, cette partition est *country='US'*.

Exécuter le Job

Vous pouvez appuyer sur la touche **F6** pour exécuter ce Job.

Cela fait, la vue **Run** s'ouvre automatiquement. Vous pouvez suivre le processus d'exécution.

Vous pouvez également vérifier les résultats dans la console Web de votre distribution Hadoop.

Si vous souhaitez plus d'informations concernant le Job, il est recommandé d'utiliser la console Web du JobTracker fournie par votre distribution Hadoop.

tHiveRow

Propriétés du tHiveRow

Famille de composant	Big Data/Hive	
Fonction	**tHiveRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes Hive QL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie. Ce composant peut également effectuer des requêtes dans une case de données HBase, une fois la case **Store by HBase** cochée.	
Objectif	Selon la nature de la requête et de la base de données, **tHiveRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
Version	*Distribution*	Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique. • Si disponible dans la liste de **Distribution**, l'option **Microsoft HD Insight** vous permet d'utiliser un cluster Microsoft HD Insight. Dans cette optique, vous devez configurez les connexions aux services WebHCat, HD Insight et Windows Azure Storage du cluster dans les zones affichées. Une vidéo démontrant comment configurer cette connexion est disponible à l'adresse suivante : https://www.youtube.com/watch?v=A3QTT6VsNoM. • L'option **Custom** vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**. Pour vous connecter à une distribution personnalisée, une fois l'option **Custom** sélectionnée, cliquez sur le bouton ... pour afficher une fenêtre, dans laquelle vous pouvez : 1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

		2. Sélectionner **Import from zip** pour importer un fichier .zip contenant une distribution personnalisée que vous pouvez télécharger depuis http://www.talendforge.org/exchange/index.php. Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d'importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant. Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez *Connexion à une distribution Hadoop personnalisée*.
	Hive version	Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez. En plus de l'évolution de Hadoop, veuillez noter les changements suivants : • Si vous utilisez **Hortonworks Data Platform V2.2**, les fichiers de configuration de votre cluster peuvent utiliser des variables d'environnement comme *${hdp.version}*. Si vous êtes dans cette situation, vous devez configurer la propriété *mapreduce.application.framework.path* dans la table **Hadoop properties** de ce composant, avec la valeur du chemin d'accès pointant explicitement vers l'archive du framework MapReduce de votre cluster. Par exemple : `mapreduce.application.framework.path=/hdp/apps/2.2.0.0-2041/mapreduce/mapreduce.tar.gz#mr-framework` • Si vous utilisez **Hortonworks Data Platform V2.0.0**, le type de système d'exploitation utilisé pour la distribution Hadoop et pour le Job **Talend** doit être identique, par exemple Windows ou Linux. Sinon, vous devez utiliser le Jobserver de **Talend** afin d'exécuter un Job dans le même type de système d'exploitation que celui dans lequel la distribution **Hortonworks Data Platform V2.0.0** utilisée est exécutée. Pour plus d'informations concernant le Jobserver de **Talend**, consultez le *Guide d'installation et de migration Talend*.
Connection	*Connection mode*	Sélectionnez un mode de connexion dans la liste. Les options varient selon la distribution que vous utilisez.
	Hive server	Sélectionnez le serveur Hive sur lequel vous souhaitez que le Job utilisant ce composant exécute des requêtes dans Hive. La liste **Hive server** est disponible uniquement lorsque la distribution Hadoop à utiliser, par exemple **HortonWorks Data Platform V1.2.0 (Bimota)** supporte HiveServer2. Vous pouvez sélectionner HiveServer2 (**Hive 2**), le serveur supportant mieux les connexions simultanées de différents clients que HiveServer (**Hive 1**). Pour plus d'informations concernant HiveServer2, consultez la page https://cwiki.apache.org/confluence/display/Hive/Setting+Up+HiveServer2 (en anglais).
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données. Ce champ n'est pas disponible lorsque vous sélectionnez **Embedded** dans la liste **Connection mode**.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Authentication	*Use kerberos authentication*	Si vous accédez au Metastore de Hive avec une sécurité Kerberos, cochez cette case et saisissez ensuite les paramètres appropriés dans les champs qui s'affichent. Les valeurs des paramètres suivants peuvent être trouvées dans le fichier *hive-site.xml* du système Hive utilisé. 1. **Hive principal** utilise la valeur de *hive.metastore.kerberos.principal*. C'est le principal du service du Metastore de Hive. 2. **Metastore URL** utilise la valeur de *javax.jdo.option.ConnectionURL*. C'est la chaîne JDBC de connexion au Metastore de Hive.

		3. **Driver class** utilise la valeur de *javax.jdo.option.ConnectionDriverName*. C'est le nom du pilote de la connexion JDBC. 4. **Username** utilise la valeur de *javax.jdo.option.ConnectionUserName*. Ce paramètre, ainsi que le paramètre **Password**, sont utilisés pour les informations de connexion de l'utilisateur au Metastore de Hive. 5. **Password** utilise la valeur de *javax.jdo.option.ConnectionPassword*. Cette case est disponible ou indisponible selon la distribution d'Hadoop à laquelle vous vous connectez.
	Use SSL encryption	Cochez cette case pour activer la connexion cryptée SSL. Les champs qui s'affichent ensuite fournissent les informations d'authentification : • Dans le champ **Trust store path**, saisissez le chemin ou parcourez votre système jusqu'au fichier TrustStore à utiliser. Par défaut, les types TrustStore supportés sont *JKS* et *PKCS 12*. • Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Cette fonctionnalité n'est disponible que pour HiveServer2 en mode **Standalone** pour les distributions suivantes : • Hortonworks Data Platform 2.0 + • Cloudera CDH4 + • Pivotal HD 2.0 +
	Use a keytab to authenticate	Cochez la case **Use a keytab to authenticate** pour vous connecter à un système Hadoop utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**. Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.
Hadoop properties	*Set Jobtracker URI*	Cochez cette case pour indiquer l'emplacement du service JobTracker à utiliser, dans le cluster Hadoop. Par exemple, si vous avez choisi une machine nommée *machine1* comme JobTracker, définissez son emplacement comme suit *machine1:portnumber*. Notez qu'un JobTracker est le service qui assigne des tâches Map/Reduce à des nœuds spécifiques dans un cluster Hadoop. Le mot "Job" dans JobTracker ne désigne pas un Job **Talend** mais un job Hadoop, décrit comme job MR ou MapReduce dans la documentation de Apache Hadoop : http://hadoop.apache.org (en anglais). Cette propriété est requise lorsque la requête que vous souhaitez utiliser est exécutée dans Windows et est une requête `Select`. Par exemple, `SELECT your_column_name FROM your_table_name` Si vous utilisez YARN dans votre cluster Hadoop, comme **Hortonworks Data Platform V2.0.0** ou **Cloudera CDH4.3 + (mode YARN)**, vous devez spécifier l'emplacement du **Resource Manager** et non celui du Jobtracker. Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d'un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du runtime) : 1. Cochez la case **Set resourcemanager scheduler address** et saisissez l'adresse de l'ordonnanceur (Scheduler) dans le champ qui apparaît. 2. Allouez des volumes de mémoire corrects aux calculs **Map** et **Reduce** ainsi qu'à l'application **ApplicationMaster** de YARN en cochant la case **Set memory** dans la vue **Advanced settings**.

		3. Cochez la case **Set jobhistory address** et saisissez l'emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory. 4. Cochez la case **Set staging directory** et saisissez le chemin d'accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l'exécution de programmes. Ce répertoire se trouve sous la propriété *yarn.app.mapreduce.am.staging-dir* dans les fichiers de configuration, notamment les fichiers *yarn-site.xml* et *mapred-site.xml* de votre distribution. 5. Cochez la case **Set Hadoop user** et saisissez le nom de l'utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d'écriture, ce champ vous permet d'exécuter le Job directement avec l'utilisateur ayant les droits d'accès appropriés au fichier ou répertoire à traiter. 6. Cochez la case **Use datanode hostname** pour permettre au Job d'accéder aux nœuds de données via leurs noms d'hôte. Cela définit la propriété *dfs.client.use.datanode.hostname* en *true*. Pour plus d'informations concernant le framework Hadoop Map/Reduce, consultez le tutoriel Map/Reduce dans la documentation de Apache Hadoop : http://hadoop.apache.org (en anglais).
	Set NameNode URI	Cochez cette case pour indiquer l'emplacement du NameNode du cluster Hadoop à utiliser. Le NameNode est le nœud maître d'un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée *masternode* comme NameNode d'une distribution Apache Hadoop, son emplacement est *hdfs://masternode:portnumber*. Cette propriété est requise lorsque la requête que vous souhaitez utiliser est exécutée dans Windows et est une requête `Select`. Par exemple, `SELECT your_column_name FROM your_table_name` Pour plus d'informations concernant le framework Hadoop Map/Reduce, consultez le tutoriel Map/Reduce dans la documentation de Apache Hadoop : http://hadoop.apache.org (en anglais).
Microsoft HD Insight properties	*WebHCat configuration*	Saisissez l'adresse et les informations d'authentification du service WebHCat du cluster Microsoft HD Insight à utiliser. Le studio utilise ce service pour soumettre le Job au cluster HD Insight. Dans le champ **Job result folder**, saisissez l'emplacement où vous souhaitez stocker les résultats d'exécution du Job dans Azure Storage.
	HDInsight configuration	Saisissez les informations d'authentification information du cluster HD Insight à utiliser.
	Windows Azure Storage configuration	Saisissez l'adresse et les informations d'authentification du compte Azure Storage à utiliser. Dans le champ **Container**, saisissez le nom du conteneur à utiliser. Dans le champ **Deployment Blob**, saisissez l'emplacement où vous souhaitez stocker le Job et ses bibliothèques dépendantes dans le compte Azure Storage.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Table Name	Nom de la table à traiter.
	Query type	Peut être **Built-in** ou **Repository**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	This query uses Parquet objects	Lorsqu'elle est disponible, cochez cette case pour indiquer que la table à gérer utilise le format Parquet et permet au composant d'appeler le fichier Jar requis. Notez que, lorsque le format de fichier à utiliser est **PARQUET**, il est possible qu'il vous soit demandé de trouver le fichier Jar Parquet spécifique et l'installer dans le studio. • Lorsque le mode de connexion à Hive est **Embedded**, le Job est exécuté sur votre machine locale et appelle ce Jar installé dans le studio. • Lorsque le mode de connexion à Hive est **Standalone**, le Job est exécuté sur le serveur hébergeant Hive et ce fichier Jar est envoyé au système HDFS du cluster auquel vous vous connectez. Assurez-vous d'avoir correctement défini l'URI du NameNode dans le champ correspondant de la vue **Basic settings**. Ce fichier Jar est téléchargeable depuis le site Web d'Apache. Pour plus d'informations concernant l'installation d'un fichier externe, consultez https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais).
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma. Pour plus d'informations concernant le langage de requêtes Hive, consultez https://cwiki.apache.org/confluence/display/Hive/LanguageManual (en anglais). Les données compressées en Gzip ou Bzip2 peuvent être traitées via les instructions de requête. Pour plus d'informations, consultez la page https://cwiki.apache.org/confluence/display/Hive/CompressedStorage (en anglais). Hadoop fournit différents formats de compression permettant de réduire l'espace nécessaire au stockage des fichiers et d'accélérer le transfert de données. Lorsque vous lisez un fichier compressé, le *Studio Talend* doit le décompresser avant de pouvoir en alimente le flux d'entrée.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
HBase Configuration Disponible uniquement lorsque la case **Use an existing connection** est décochée.	*Store by HBase*	Cochez cette case afin d'afficher les paramètres à configurer pour permettre aux composants Hive d'accéder aux tables HBase : • Une fois l'accès configuré, vous pourrez utiliser, dans un **tHiveRow** et un **tHiveInput**, les instructions Hive QL permettant de lire et d'écrire des données dans HBase. • Si vous utilisez l'authentification Kerberos, vous devez définir les principaux relatifs à HBase dans les champs correspondants qui sont affichés. Pour plus d'informations à propos de cet accès concernant Hive et HBase, consultez la documentation de Apache Hive concernant l'intégration Hive/HBase.
	Zookeeper quorum	Saisissez le nom ou l'URL du service Zookeeper utilisé pour coordonner les transactions entre **Talend** et HBase. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer la propriété **zookeeper.znode.parent** pour définir la racine du chemin relatif d'un fichier HBase Zookeeper puis de sélectionner la case **Set Zookeeper znode parent** afin de définir cette propriété.
	Zookeeper client port	Saisissez le numéro du port d'écoute client du service Zookeeper que vous utilisez.

	Define the jars to register for HBase	Cochez cette case pour afficher la table **Register jar for HBase**, dans laquelle vous pouvez enregistrer tout fichier Jar manquant, requis pour HBase, par exemple, Hive Storage Handler, enregistré par défaut avec votre installation Hive.
	Register jar for HBase	Cliquez sur le bouton pour ajouter des lignes à la table, puis, dans la colonne **Jar name**, sélectionnez le(s) fichier(s) Jar à enregistrer. Dans la colonne **Jar path**, saisissez le chemin d'accès à ce(s) Jar(s).
Advanced settings	*Temporary path*	Si vous ne souhaitez pas configurer le Jobtracker et le NameNode lorsque vous exécutez la requête `select * from your_table_name`, vous devez paramétrer un chemin d'accès temporaire. Par exemple, */C:/select_all* sous Windows.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Hadoop properties	Le *Studio Talend* utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l'exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut. • Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**. Pour plus d'informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation Apache Hadoop sur le site http://hadoop.apache.org (en anglais) ou la documentation de la distribution Hadoop que vous utilisez. A titre d'exemple, les liens vers certaines propriétés sont listés ci-après: • Généralement, les propriétés relatives à HDFS peuvent être trouvées dans le fichier *hdfs-default.xml* correspondant à votre distribution, comme par exemple http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais). • Apache fournit également une page listant toutes les propriétés relatives à Hive : http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais).
	Hive properties	Le *Studio Talend* utilise la configuration par défaut pour son moteur afin d'effectuer des opération dans un base de données Hive. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez cette table avec la (les) propriété(s) à personnaliser. Ensuite, à l'exécution, la (les) propriété(s) personnalisée(s) écrasent celles par défaut. Pour plus d'informations concernant les propriétés dédiées à Hive, consultez https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration (en anglais). • Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.
	Mapred job map memory mb et *Mapred job reduce memory mb*	Si le système Hadoop utilisé est Hortonworks Data Platform V1.2 ou Hortonworks Data Platform V1.3, vous devez définir des allocations de mémoire suffisantes pour que les opérations de map et reduce soient effectuées par le système Hadoop. Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations *map* et *reduce* dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux *1000*, ce qui est normalement adapté pour l'exécution de ces opérations.
	Path separator in server	Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d'autres termes, changez le séparateur si celui-ci n'est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.

	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes Hive QL. Si le *Studio Talend* utilisé pour vous connecter à une base de données Hive fonctionne sous Windows, vous devez créer manuellement un dossier appelé *tmp* à la racine du disque où le *Studio Talend* est installé.	
Prérequis	La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d'exemple relatives à MapR. • Assurez-vous d'avoir installé le client MapR sur la même machine que le *Studio Talend* et d'avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D'après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées *MAPR_INSTALL\ hadoop\hadoop-VERSION\lib\native*. Par exemple, pour Windows, la bibliothèque est *lib\MapRClient.dll* dans le fichier Jar du client MapR. Pour plus d'informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n'ajoutez pas de librairie, il est possible que vous rencontriez l'erreur suivante : `no MapRClient in java.library.path`. • Configurez l'argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]**. Cet argument fournit au studio le chemin d'accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d'utiliser entièrement l'aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR. Pour plus d'informations concernant la configuration de cet argument, consultez la section expliquant comment visualiser les données dans le *Guide de prise en main de Talend Open Studio for Big Data*. Pour plus d'informations concernant l'installation d'une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

Lors de la configuration du composant devant se connecter à une distribution Hadoop, gardez à l'esprit que des paramètres sont requis par Hadoop, tels que le NameNode et le Jobtracker.

Composants Business

Ce chapitre passe en revue les principaux composants présents dans la famille **Business** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **Business** regroupe les connecteurs couvrant des besoins spécifiques, tels que la lecture et l'écriture de base de données de type CRM ou ERP et la lecture et l'écriture de système SAP.

tAlfrescoOutput

Propriétés du tAlfrescoOutput

Famille du composant	Business	
Fonction	Le composant **tAlfrescoOutput** permet de créer des documents dématérialisés sur un serveur Alfresco où ils sont indexés selon des modèles significatifs.	
Objectif	Ce composant permet de créer et de gérer des documents sur un serveur Alfresco.	
Basic settings	*URL*	Saisissez l'URL pour vous connecter à l'application Web Alfresco.
	Login et *Password*	Entrez les données d'identification de l'utilisateur au serveur Alfresco. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Target Location	*Base*	Saisissez le chemin d'accès de l'emplacement où stocker le document, ou Cochez la case **Map...** puis choisissez, à partir de la liste déroulante **Column**, l'emplacement d'arrivée. Lorsque vous saisissez le nom de l'emplacement, assurez-vous de le protéger par des doubles barres obliques inversées (\\).
Create Or Update Mode	*Document Mode*	Sélectionnez, à partir de la liste déroulante, le mode que vous voulez utiliser pour créer votre document. **Create only** : permet de créer un document s'il n'existe pas préalablement. Notez cependant qu'un message d'erreur apparaîtra si vous essayez de créer un document qui existe déjà. **Create or update** : permet de créer un document qui n'existe pas encore ou de mettre à jour un document existant.
	Container Mode	Sélectionnez, à partir de la liste déroulante, le mode que vous voulez utiliser pour le dossier de destination sur le serveur Alfresco. **Update only** : permet de mettre à jour un dossier de destination déjà existant. Notez cependant qu'un message d'erreur apparaîtra si vous essayez de mettre à jour un document qui n'existe pas. **Create or update** : permet de créer un dossier de destination s'il n'existe pas au préalable, ou de mettre à jour un dossier existant.
	Define Document Type	Cliquez sur les trois points **[...]** pour afficher l'éditeur **tAlfrescoOutput**. Cet éditeur vous permet : - de sélectionner le fichier dans lequel vous avez défini les métadonnées pour servir de cadre lors de l'enregistrement du document sur Alfresco - de définir le type du document

		- de sélectionner, à partir de la liste déroulante **available aspects**, l'un des aspects du modèle, puis de cliquer sur le bouton **[+]** pour ajouter cet aspect à la liste de gauche.
	Property Mapping	Affiche les paramètres renseignés dans l'éditeur **tAlfrescoOutput**, selon lesquels le document sera créé sur le serveur Alfresco. Notez que vous pouvez modifier n'importe lequel des schémas d'entrée de la zone **Property Mapping**.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Result Log File Name	Permet de parcourir le fichier dans lequel vous voulez enregistrer les logs liés à l'exécution du Job.
	Die on error	Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur. Si nécessaire, vous pouvez récupérer les lignes en erreur via un lien **Row** > **Rejects**.
Advanced settings	*Configure Target Location Container*	Permet de configurer le type par défaut des conteneurs. Cochez cette case pour afficher de nouveaux champs dans lesquels vous pouvez modifier le type du conteneur et utiliser les types que vous avez créés sur le modèle père/fils.
Permissions	*Configure Permissions*	Lorsque cette case est cochée, l'option permet de configurer manuellement les droits d'accès aux conteneurs et aux documents. Cochez la case **Inherit Permissions** pour synchroniser les droits d'accès entre les conteneurs et les documents. Cliquez sur le bouton **[+]** pour ajouter de nouvelles lignes à la liste **Permissions** ; vous pourrez alors assigner des rôles à un utilisateur ou groupe d'utilisateurs dans les colonnes **User or group** et **User or group column**.
	Encoding	Sélectionnez le type d'encodage à partir de la liste ou choisissez l'option **Custom** pour le définir manuellement. Ce champ doit obligatoirement être renseigné.
	Association Target Mapping	Permet de créer de nouveaux documents sur Alfresco qui contiennent des liens associés à des documents déjà existants sur ce serveur, pour faciliter le processus de navigation par exemple. Pour créer des associations : 1. Ouvrez l'éditeur **tAlfresco**.

		2. Cliquez sur le bouton **Add** puis sélectionnez un modèle pour lequel vous avez déjà défini des aspects contenant des associations. 3. Cliquez sur la liste déroulante, en haut de la fenêtre de l'éditeur, et sélectionnez le type de document correspondant. 4. Cliquez sur **OK** pour fermer l'éditeur et afficher l'association ainsi créée dans la liste **Association Target Mapping**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de sortie et requiert un composant de début.	
Limitation/prérequis	Pour pouvoir utiliser le composant **tAlfrescoOutput**, certaines ressources doivent être installées. Pour plus d'informations, consultez la sous-section **Procédure d'installation** ci-dessous. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Procédure d'installation

Pour pouvoir utiliser le composant **tAlfrescoOutput** dans la perspective **Integration** du *Studio Talend*, vous devez tout d'abord installer le serveur Alfresco ainsi que quelques ressources utiles.

Les sous-sections ci-après indiquent en détail les conditions préalables nécessaires, ainsi que la procédure d'installation.

Conditions préalables

Commencez par exécuter les opérations suivantes :

1. Téléchargez le fichier `alfresco-community-tomcat-2.1.0.zip`
2. Dézippez-le dans un dossier d'installation, par exemple : `C:\Program Files\Java\jdk1.6.0_27`
3. Installez JDK 1.6.0+

4. Effectuez une mise à jour de la variable `JAVA_HOME (JAVA_HOME= C:\alfresco)`

5. A partir du dossier d'installation (`C:\alfresco`), lancez le serveur Alfresco en utilisant le script `alf_start.bat`

*Assurez-vous que le serveur Alfresco est bien lancé avant d'utiliser le composant **tAlfrescoOutput**.*

Installation du module Talend Alfresco

Notez que le fichier `talendalfresco_20081014.zip` est proposé avec le composant **tAlfrescoOutput** dans la perspective **Integration** du *Studio Talend*.

Pour installer le module Talend Alfresco :

1. A partir du fichier `talendalfresco_20081014.zip`, dans le dossier `talendalfresco_20081014\alfresco`, recherchez les fichiers JAR suivants : `stax-api-1.0.1.jar`, `wstx-lgpl-3.2.7.jar`, `talendalfresco-client_1.0.jar` et `talendalfresco-alfresco_1.0.jar`, puis déplacez-les vers `C:\alfresco\tomcat\webapps\alfresco\WEB-INF\lib`

2. Ajoutez le filtre d'identification des commandes au fichier `web.xml` situé au chemin d'accès `C:\alfresco\tomcat\webapps\alfresco\WEB-INF son WEB-INF/`, d'après le modèle de l'exemple situé dans le dossier `talendalfresco_20081014/alfresco` du fichier zippé `talendalfresco_20081014.zip`

   Les capture d'écran ci-dessous montrent la partie à ajouter (lignes en bleu) au fichier `web.xml alfresco.`

```
<filter>
   <filter-name>Admin Authen
   <filter-class>org.alfresc
</filter>

<filter-mapping>
   <filter-name>Authenticati
   <url-pattern>/faces/*</ur
</filter-mapping>
```

```
<filter>
   <filter-name>Admin Authentication Filter</filter-name>
   <filter-class>org.alfresco.web.app.servlet.AdminAuthenticationFilter</filter-class>
</filter>

<!-- [talendalfresco] command servlet auth filter def -->
<filter>
   <filter-name>Command Authentication Filter</filter-name>
   <filter-class>fr.openwide.talendalfresco.rest.server.CommandAuthenticationFilter</filter-class>
</filter>

<filter-mapping>
   <filter-name>Authentication Filter</filter-name>
   <url-pattern>/faces/*</url-pattern>
</filter-mapping>
```

```
<filter-mapping>
   <filter-name>Admin Authen
   <url-pattern>/faces/jsp/d
</filter-mapping>

<listener>
   <listener-class>org.apach
</listener>
```

```
<filter-mapping>
   <filter-name>Admin Authentication Filter</filter-name>
   <url-pattern>/faces/jsp/dialog/system-info.jsp</url-pattern>
</filter-mapping>

<!-- [talendalfresco] command servlet auth filter mapping -->
<filter-mapping>
   <filter-name>Command Authentication Filter</filter-name>
   <url-pattern>/command/*</url-pattern>
</filter-mapping>

<listener>
   <listener-class>org.apache.myfaces.webapp.StartupServletContextList
</listener>
```

Informations utiles pour une utilisation avancée du composant

Installer de nouveaux types pour Alfresco :

A partir du fichier `package_jeu_test.zip` , dans le dossier `package_jeu_test/fichiers_conf_alfresco2.1`, recherchez les fichiers suivants : `xml H76ModelCustom.xml` (description du modèle), `web-client-config-custom.xml` (interface Web du modèle) et `custom-model-context.xml` (enregistrement du nouveau modèle), puis collez-les dans le dossier suivant : `C:/alfresco/tomcat/shared/classes/alfresco/extension`

Dates :

- Les dates doivent correspondre au type **Talend**, `java.util.Date`
- Les colonnes qui ne comportent ni mapping, ni valeur par défaut (celles du type `Date` par exemple) se voient attribuer une chaîne de caractères vide.

  Solution : effacez toutes les colonnes sans mapping ni valeur par défaut. Notez cependant qu'une modification de type Alfresco les rétablira.

Contenu :

- Ne confondez pas le chemin d'accès au fichier dont vous voulez créer le contenu et celui de son emplacement cible sur Alfresco.
- N'oubliez pas l'URL. Elle permet de cibler de nombreux protocoles, dont le transfert de fichiers et le HTTP.
- Précédez les URLs renvoyant à des fichiers du système de fichiers en réseau de "file:", si vous utilisez Windows en local, ou de "file://", si vous utilisez soit Windows sur un réseau (qui accepte également la mention "file: \ \"), soit Linux.
- N'utilisez jamais la double barre oblique inversée (\\) dans le chemin d'accès cible (échappement automatique), à moins que vous n'entriez le chemin directement dans les propriétés de base (**Basic settings**) du composant **tAlfrescoOutput** ou que vous n'effectuiez une concaténation, dans l'éditeur **tMap** par exemple.

Propriétés multiples ou associations :

- Il est possible de n'en créer qu'une seule par document en y mappant une valeur String, ou une ou plusieurs en y mappant une valeur List(objet).
- Il est possible de vider une propriété multiple en lui mappant une liste vide que vous pouvez créer, par exemple, en utilisant la commande `new java.util.ArrayList()`dans le composant **tMap**.

En revanche, il est impossible de supprimer une association.

Faire des `List(object)`avec le composant **tAggregate** :

- définissez dans un fichier une table de relation n-n, contenant par exemple une ligne `name` (pouvant être retrouvée dans les input rows) et une ligne `category` (pouvant être définie avec son mapping dans un fichier tiers).
- group by : input name, output name.
- opération : output `categoryList`, fonction `list(object)`, input `category`. ATTENTION il s'agit d'une list (object) et non d'une simple list !

Références (documents et dossiers) :

- Les références sont créées par mapping d'une ou plusieurs références de nœud existantes (xpath ou namepath), en type `String` ou `List(object)`.
- Une erreur dans l'association ou la propriété de type référence n'empêche pas la création du nœud qui la porte.
- Les propriétés de type référence sont créées dans l'onglet des propriétés standards (**Basic settings**).
- Les associations sont créées dans l'onglet des propriétés avancées (**Advanced settings**).

Dématérialisation, tAlfrescoOutput et Gestion de Contenu

La dématérialisation est le processus qui sert à convertir les documents concrets au format numérique, permettant ainsi de passer d'une utilisation de documentation physique à des systèmes de Gestion de Contenu électroniques.

Un système de Gestion de Contenu permet de gérer un large éventail de documents, allant des plus basiques aux certificats d'actions par exemple.

Les entreprises procèdent à la dématérialisation de leur contenu par traitement des documents, qui peut être réalisé soit à la main, soit de manière automatique grâce à une machine

Compte tenu de la diversité des contenus à dématérialiser, les entreprises sont obligées d'utiliser des technologies diverses. Scanner les documents papier, créer des interfaces pour capter les documents électroniques d'autres applications, convertir les documents images en documents textes lisibles et modifiables par une machine... Ceci n'est qu'un exemple des technologies employées.

De plus, les documents scannés ainsi que les fax numériques ne peuvent être lus par des machines. Différentes technologies de reconnaissance des caractères sont utilisées pour les convertir en données utilisables. La Reconnaissance de Caractères Manuscrits (RCM) et la Reconnaissance Optique de Marques (ROM) en sont deux exemples.

Les métadonnées de support, qui permettent d'identifier le contenu de manière efficace grâce à des requêtes spécifiques, sont au moins aussi importantes que le contenu capté lors de la dématérialisation, en différents formats et à partir de nombreuses sources.

Dès lors, comment agréger et indexer dans un système de Gestion de Contenu le contenu d'un document en même temps que ses métadonnées associées, pour qu'il puisse être extrait et traité de manière significative ? **Talend** vous apporte la réponse par le biais de son composant **tAlfrescoOutput**.

Le composant **tAlfrescoOutput** vous permet de stocker et de gérer vos documents électroniques, ainsi que les métadonnées qui y sont associées, sur le serveur Alfresco, système leader sur le marché de la Gestion de Contenu.

La figure ci-dessous illustre le rôle de **Talend** à la fois dans le processus de dématérialisation et dans le système Gestion de Contenu (Alfresco).

ECMS: Enterprise Content Management System
HCR: Handwritten Character Recognition
OMR: Optical Mark Recognition

Scénario : Créer des documents sur un serveur Alfresco

Ce scénario Java décrit un Job à deux composants permettant de créer deux fichiers de documents et leurs métadonnées associées sur un serveur Alfresco, le système de Gestion de Contenu en Java.

Configurer votre Job

1. A partir de la **Palette**, cliquez-déposez les composants **tFileInputDelimited** et **tAlfrescoOutput** sur l'éditeur graphique.

2. Reliez ces deux composant à l'aide d'un lien **Row Main**.

Configurer le schéma

1. Dans l'éditeur graphique, double-cliquez sur le **tFileInputDelimited** pour en afficher les paramètres de base (**Basic settings**).

2. Renseignez le chemin d'accès au fichier dans le champ **File Name** ainsi que toutes les autres propriétés. Notez cependant que si vous avez déjà stocké vos schémas d'entrée en local dans le **Repository**, vous pouvez tout simplement récupérer le fichier choisi à partir du nœud **Metadata** et le déposer sur l'éditeur graphique. Les paramètres du fichier délimité s'afficheront alors automatiquement dans les champs concernés de l'onglet **Basic settings** du composant concerné.

   Pour plus d'informations concernant les métadonnées, consultez le *Guide utilisateur du Studio Talend*.

   Dans ce scénario, le fichier délimité présente les métadonnées et le chemin d'accès de deux documents que vous allez créer sur le serveur Alfresco. Le schéma d'entrée des documents de décompose en quatre colonnes : `file_name`, `destination_folder name`, `source_path`, et `author`.

	A	B	C	D	E	F
1	file_name;destination_folder_name;source_path;author					
2	PO_43278.pdf;FINANCE\PO;file:D:/PO/customer1_2009-03-24.pdf;talend					
3	PO_43279.pdf;FINANCE\PO;file:D:/PO/customer2_2009-03-24.pdf;talend					

   Le schéma d'entrée du fichier délimité sera donc comme suit :

Configurer la connexion au serveur Alfresco

1. Dans l'éditeur graphique, double-cliquez sur le composant **tAlfrescoOutput** pour en afficher les paramètres de base (**Basic settings**).

2. Dans la partie **Alfresco Server**, saisissez l'URL du serveur Alfresco ainsi que les données d'identification de l'utilisateur dans les champs correspondants.

3. Dans la partie **TargetLocation**, vous pouvez soit saisir le nom de l'emplacement du serveur où vous voulez stocker le document, soit cocher la case **Map...** puis choisir l'emplacement cible à partir de la liste déroulante **Column**, soit `destination_folder_name` dans ce scénario.

Lorsque vous renseignez le nom de l'emplacement, assurez vous de bien utiliser le caractère d'échappement double barre oblique inversée (\\).

4. Dans la liste déroulante **Document Mode**, sélectionnez le mode que vous voulez utiliser pour la création de vos documents.

5. Dans la liste déroulante **Container Mode**, sélectionnez le mode que vous voulez utiliser pour le dossier de destination sur Alfresco.

Définir le document

1. Cliquez sur le bouton **[...]** de l'option **Define Document Type** pour ouvrir l'éditeur **tAlfrescoOutput**.

2. Cliquez sur le bouton **Add** pour parcourir et sélectionner le fichier xml qui comprend les métadonnées selon lesquelles vous voulez sauvegarder les documents sur Alfresco. Tous les aspects disponibles du modèle sélectionné s'affichent dans la liste **Available Aspects**.

Vous pouvez parcourir le dossier modèle en local ou en réseau. Une fois définis les aspects à appliquer au document que vous allez créer sur Alfresco, vous n'aurez plus besoin de ce dossier.

3. Si nécessaire, sélectionnez le ou les aspect(s) à ajouter aux métadonnées à écrire sur le serveur Alfresco, à partir de la liste déroulante **Available Aspects**. Dans ce scénario, l'objectif est de faire en sorte que le nom de l'auteur fasse partie des métadonnées enregistrées sur Alfresco

4. Cliquez sur la liste déroulante, située en haut de la fenêtre de l'éditeur, pour sélectionner le type à donner aux documents créés sur Alfresco. Dans cet exemple, il s'agit de `Content`.

Tous les aspects définis, utilisés pour sélectionner les métadonnées à écrire sur Alfresco, s'affichent dans la liste **Property Mapping** de l'onglet **Basic settings** du composant **tAlfrescoOutput**. Dans ce scénario, il s'agit de trois aspects, deux aspects de base pour le type `Content` (`content` et `name`) ainsi qu'un aspect additionnel (`author`).

Exécuter votre Job

1. Cliquez sur **Sync columns** pour propager automatiquement toutes les colonnes du fichier délimité.

2. Si nécessaire, cliquez sur **Edit schema** pour avoir un aperçu de la structure des données du **tAlfrescoOutput** en sortie.

3. Cliquez sur le bouton **[...]** situé à côté du champ **Result Log File Name** puis parcourez le fichier où vous voulez sauvegarder les logs après exécution du Job.

4. Enregistrez votre Job puis appuyez sur **F6** pour l'exécuter.

Les deux documents sont créés sur Alfresco grâce aux métadonnées des schémas d'entrée.

Accueil Mon accueil Espace Invité Mon Alfresco
Remonter un dysfonctionnement
Déconnexion
Navigateur
Mon Alfresco
Accueil
Mon accueil
Espace Invité
Mon Alfresco > FINANCE > PO
PO
Cette vue permet de parcourir les éléments de cet Espace.
(0)
Ajouter un contenu
Créer
Parcourir les Espaces
Aucun élément à afficher. Cliquer sur 'Créer un Espace' pour créer un espace.
Page 1 sur 1
Eléments de Contenu
PO_43278.pdf
58,14 Ko
13 février 2009 15:47
PO_43279.pdf
58,14 Ko
13 février 2009 15:47
PO_43278.pdf
Created: 13 févr. 2009 15:47:38
Creator: admin
Modified: 13 févr. 2009 15:47:38
Modifier: admin
Author: talend
Size: 59,54 KB
Preview:
1 - Ordre d'installation des modules
square4 JVM 1.5 ou + par SUN
square4 d'un SGBD (MySQL ou SQL SERVER)
Alfresco Community

tBonitaDeploy

Propriétés du tBonitaDeploy

Famille de composant	Business/Bonita	
Fonction	Le **tBonitaDeploy** configure un moteur Bonita Runtime et déploie un processus spécifique Bonita (un fichier .bar exporté de la solution Bonita solution) vers ce moteur.	
Objectif	Ce composant déploie un processus spécifique Bonita vers Bonita Runtime.	
Basic settings	*Bonita version*	Sélectionnez un numéro de version pour le moteur Bonita Runtime.
Bonita Runtime Configuration	*Bonita Runtime Environment File*	Parcourez votre répertoire, ou saisissez le chemin d'accès au fichier d'environnement Bonita Runtime. Ce champ est affiché uniquement lorsque vous sélectionnez la version 5.3.1 de Bonita dans la liste **Bonita version**.
	Bonita Runtime Home	Parcourez votre système jusqu'au répertoire d'environnement de Bonita Runtime, ou saisissez son chemin d'accès. Ce champ est affiché uniquement lorsque vous sélectionnez la version 5.6.1 de Bonita dans la liste **Bonita version**.
	Bonita Runtime Jass File	Parcourez votre répertoire, ou saisissez le chemin d'accès au fichier jass Bonita Runtime.
	Bonita Runtime logging file	Parcourez votre répertoire, ou saisissez le chemin d'accès au fichier de log Bonita Runtime.
	Login Module	Saisissez le nom du module de login au moteur Bonita Runtime défini dans le fichier Bonita Runtime jaas.
	Business Archive	Parcourez votre répertoire, ou saisissez le chemin d'accès au fichier .bar Bonita de traitement que vous souhaitez utiliser.
	User name	Saisissez votre identifiant de connexion à Bonita.
	Password	Saisissez votre mot de passe de connexion à Bonita. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Die on error	Cette case n'est pas cochée par défaut, pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu'au niveau de chaque composant.
Connections	Liens de sortie (de ce composant à un autre) : **Trigger** : Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError. Liens d'entrée (d'un autre composant à celui-ci) : **Trigger** : Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError Pour plus d'informations concernant les connexions, consultez la section relatives aux types de connexions, dans le *Guide utilisateur du Studio Talend*.	
Global Variables	**ProcessDefinitionUUID** : identifiant du processus déployé. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en standalone. Ce composant fonctionne en étroite relation avec les éléments BPM. Vous devez souscrire à l'une des solutions **Talend** avec BPM pour utiliser ce composant efficacement. Sinon, téléchargez manuellement la solution Bonita à utiliser.
Limitation	Le fichier d'environnement Bonita Runtime, le fichier jaas Bonita Runtime et le fichier de log Bonita Runtime doivent être stockés sur le serveur d'exécution du Job, à l'aide de ce composant.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tBonitaInstantiateProcess

Propriétés du tBonitaInstantiateProcess

Famille de composant	Business/Bonita	
Fonction	Le composant **tBonitaInstantiateProcess** instancie un processus existant dans un moteur Bonita Runtime puis lance l'exécution.	
Objectif	Ce composant commence l'instance d'un processus spécifique déployé dans le moteur Bonita Runtime.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur **Edit schema** afin de visualiser le schéma. Pour ce composant, le schéma dépend du **Module** sélectionné. La colonne ProcessInstanceUUID est prédéfinie dans le schéma de ce composant et réservée à l'identifiant de l'instance du processus créée
	Bonita Client Mode	Sélectionnez le mode client que vous souhaitez utiliser pour instancier un processus BPM. Pour plus d'informations concernant les modes client de Bonita, consultez les manuels de Bonita.
	URL	Saisissez l'URL du serveur d'application Web Bonita auquel vous devez accéder pour l'instanciation du processus. Ce champ est disponible uniquement en mode HTTP client.
Authorization	*Auth Username* et *Auth Password*	Saisissez les informations d'authentification pour vous connecter au serveur d'application Web Bonita en tant qu'utilisateur technique. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Les informations par défaut sont automatiquement renseignées dans ces champ. Pour plus d'informations, consultez les manuels de Bonita. Ces champs sont disponibles uniquement en mode HTTP client.
	Bonita version	Sélectionnez la version du moteur Bonita Runtime à utiliser. Ce champ est disponible uniquement en mode Java client.
	Bonita Runtime Environment File	Parcourez votre système jusqu'au fichier d'environnement de Bonita Runtime, ou saisissez son chemin d'accès. Ce champ est disponible uniquement en mode Java client. Ce champ est affiché uniquement lorsque vous sélectionnez la version 5.3.1 de Bonita dans la liste **Bonita version**.

	Bonita Runtime Home	Parcourez votre système jusqu'au répertoire d'environnement de Bonita Runtime, ou saisissez son chemin d'accès. Ce champ est affiché uniquement lorsque vous sélectionnez la version 5.6.1 de Bonita dans la liste **Bonita version**.
	Bonita Runtime Jass File	Parcourez votre répertoire, ou saisissez le chemin d'accès au fichier jaas Bonita Runtime. Ce champ est disponible uniquement en mode Java client.
	Bonita Runtime logging file	Parcourez votre répertoire, ou saisissez le chemin d'accès au fichier de log Bonita Runtime. Ce champ est disponible uniquement en mode Java client.
	Use Process ID	Cochez cette case pour instancier un processus existant. Une fois la case cochée, le champ **Process definition ID** est activé, et vous pouvez saisir l'ID du processus. Ce champ est disponible uniquement en mode Java client. L'ID du processus est créé lorsque ce processus est déployé dans le moteur Bonita Runtime.
	Process Name et *Process Version*	Saisissez les informations d'ID d'un processus spécifique que vous souhaitez instancier. Ces informations sont utilisées pour générer automatiquement l'ID du processus. Ce champ est disponible pour les modes Java client et HTTP client.
	User name	Saisissez l'identifiant utilisé pour instancier ce processus. Ce champ est disponible pour les modes Java client et HTTP client.
	Password	Saisissez le mot de passe utilisé pour instancier ce processus. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Ce champ est disponible uniquement en mode Java client.
	Die on error	Cette case n'est pas cochée par défaut, pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu'au niveau de chaque composant.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main (fournit les paramètres de sortie à partir du processus). **Trigger** : Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main. **Trigger** : Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError Pour plus d'informations à propos des connexions, consultez la section concernant les types de connexions dans le *Guide utilisateur du Studio Talend*.	
Global Variables	**ProcessInstanceUUID** : identifiant de l'instance du processus créé. Cette variable est une variable **Flow** et retourne une chaîne de caractères. L'identifiant peut également être récupéré à l'aide d'un lien de sortie **Row** > **Main**.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en standalone ou en tant que composant de sortie. Ce composant fonctionne en étroite relation avec les éléments BPM. Vous devez souscrire à l'une des solutions **Talend** avec BPM pour utiliser ce composant efficacement. Sinon, téléchargez manuellement la solution Bonita à utiliser.
Limitation	Le fichier d'environnement Bonita Runtime, le fichier jaas Bonita Runtime et le fichier de log Bonita Runtime doivent être stockés sur le serveur d'exécution du Job utilisant ce composant.

Scénario 1 : Exécuter un processus Bonita via un Job Talend

Ce scénario décrit un Job déployant un processus Bonita dans le moteur Bonita Runtime et exécutant ce processus, dans lequel une requête sur le personnel est traitée.

Ce Job utilise trois composants.

- Un **tBonitaDeploy** : ce composant déploie un processus Bonita dans le moteur Bonita Runtime.

- Un **tFixedFlowInput** : ce composant génère le schéma utilisé comme paramètre d'exécution du processus déployé.

- Un **tBonitaInstantiateProcess** : ce composant exécute le processus déployé.

Avant de commencer à reproduire ce schéma, préparez votre fichier Bonita .bar. Si vous n'avez pas souscrit à l'une des solutions Talend avec BPM, vous devez exporter manuellement ce fichier depuis le système Bonita, puis le déployer dans le moteur Bonita Runtime, à l'aide, par exemple du composant **tBonitaDeploy**, présenté dans ce scénario. Dans ce scénario, ce fichier est *TEST--4.0.bar*. Une fois déployé, ce processus peut être vérifié via l'interface Bonita.

Pour reproduire ce scénario, procédez comme suit :

Configurer le Job

1. Déposez les composants **tBonitaDeploy**, **tFixedFlowInput** et **tBonitaInstantiateProcess** de la **Palette** dans l'espace de modélisation graphique.

2. Cliquez-droit sur le composant **tBonitaDeploy** pour ouvrir son menu contextuel.

3. Sélectionnez **Trigger** > **OnSubjobOk** pour relier le **tBonitaDeploy** au **tFixedFlowInput**.

4. Cliquez-droit sur le **tFixedFlowInput** pour ouvrir son menu contextuel et sélectionnez **Row** > **Main** pour relier ce composant au **tBonitaInstantiateProcess**, à l'aide d'un lien **Main**.

Configurer le déploiement du processus

Pour reproduire ce scénario, procédez comme suit :

1. Double-cliquez sur le **tBonitaDeploy** afin d'ouvrir sa vue **Basic settings**.

2. Sélectionnez la version 5.3.1 de Bonita dans la liste **Bonita version**. La version sélectionnée doit être la même que celle du moteur Bonita Runtime que vous utilisez.

3. Dans la zone **Bonita Runtime Configuration**, parcourez votre répertoire de fichiers de variables Bonita Runtime. Dans le champ **Bonita Runtime Environment file**, parcourez votre répertoire jusqu'au

fichier *bonita-environnement.xml*. Dans le champ **Bonita Runtime Jaas File**, parcourez votre répertoire jusqu'au fichier *jaas-standard.cfg*. Dans le champ **Bonita Runtime Logging File**, parcourez votre répertoire jusqu'au fichier *logging.properties*.

Pour les utilisateurs de la version 5.2.3 de Bonita, seuls les champs **Bonita Runtime Jaas File** et **Bonita Runtime Logging File** sont à renseigner.

Pour les utilisateurs de la version 5.6.1 de Bonita, parcourez votre système afin de renseigner le champ **Bonita Runtime Home** avec le répertoire d'environnement de Bonita Runtime.

4. Dans le champ **Business Archive**, parcourez votre répertoire jusqu'au fichier Bonita .bar, qui est le processus exporté de votre système Bonita et sera déployé dans le moteur Bonita Runtime.

5. Dans les champs **Username** et **Password**, saisissez vos informations de connexion à Bonita.

Configurer le flux d'entrée

1. Double-cliquez sur le composant **tFixedFlowInput** afin d'ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

3. Cliquez sur le bouton **[+]** pour ajouter une ligne. Nommez-la *Name*.

   Le nom est identique à celui du paramètre configuré dans Bonita, afin d'exécuter le même processus. Ainsi, Bonita reconnaît la colonne comme un paramètre valide et lit sa valeur pour instancier le processus.

4. Cliquez sur **OK**.

5. Dans la zone **Mode** de la vue **Basic settings**, sélectionnez l'option **Use inline table** puis cliquez sur le bouton **[+]** afin d'ajouter une ligne.

6. Dans le tableau, cliquez sur la ligne ajoutée et saisissez entre guillemets le nom de la personne de votre personnel : *ychen*, dont la requête sera traitée par le processus déployé.

Configurer les paramètres du tBonitaInstantiateProcess

1. Double-cliquez sur le composant **tBonitaInstantiateProcess** afin d'ouvrir sa vue **Basic settings**.

2. Sélectionnez la version 5.3.1 de Bonita dans la liste **Bonita version**. La version sélectionnée doit être la même que celle du moteur Bonita Runtime que vous utilisez.

3. Dans la zone **Bonita Runtime Configuration**, parcourez votre répertoire de fichiers de variables Bonita Runtime. Dans le champ **Bonita Runtime Environment file**, parcourez votre répertoire jusqu'au fichier *bonita-environnement.xml*. Dans le champ **Bonita Runtime Jaas File**, parcourez votre répertoire jusqu'au fichier *jaas-standard.cfg*. Dans le champ **Bonita Runtime Logging File**, parcourez votre répertoire jusqu'au fichier *logging.properties*.

   Pour les utilisateurs de la version 5.2.3 de Bonita, seuls les champs **Bonita Runtime Jaas File** et **Bonita Runtime Logging File** sont à renseigner.

   Pour les utilisateurs de la version 5.6.1 de Bonita, parcourez votre système afin de renseigner le champ **Bonita Runtime Home** avec le répertoire d'environnement de Bonita Runtime.

4. Cochez la case **Use Process ID** pour activer le champ **Process Definition Id**.

5. Dans le champ **Process Definition Id**, cliquez entre les guillemets et appuyez sur **Ctrl+Espace** afin d'ouvrir la liste d'autocomplétion contenant les variables globales disponibles pour ce Job.

6. Double-cliquez sur la variable que vous souhaitez utiliser pour l'ajouter entre les guillemets. Dans ce scénario, double-cliquez sur *tBonitaDeploy_1_ProcessDefinitionUUID*, qui récupère l'ID de définition du processus déployé par le composant **tBonitaDeploy**.

   Vous pouvez également décocher la case **Use Process ID** afin d'activer les champs **Process name** et **Process version** et renseigner les informations correspondantes. Le **tBonitaInstantiateProcess** concatène le nom et la version du processus que vous avez saisis lors de la construction de l'ID du processus.

7. Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant et votre mot de passe de connexion à Bonita.

Exécuter le Job

- Appuyez sur **F6** pour exécuter le Job.

Le processus est déployé dans le moteur Bonita Runtime et une instance est créée pour les requêtes concernant le personnel.

Scénario 2 : Ecrire en sortie l'UUID de l'instance du processus via le lien Row > Main

Ce scénario déploie un processus Bonita dans Bonita Runtime, démarre une instance et écrit en sortie l'UUID de l'instance du processus via le lien **Row** > **Main**.

Relier les composants

1. Déposez de la **Palette** dans l'espace de modélisation graphique un composant **tBonitaDeploy**, un **tBonitaInstantiateProcess** et un **tLogRow**.

2. Renommez le **tBonitaDeploy** en **deploy_process**, le **tBonitaInstantiateProcess** en **start_instance** et le **tLogRow** en **show_instance_uuid**.

3. Reliez le composant **tBonitaDeploy** au **tBonitaInstantiateProcess** à l'aide d'un lien **OnSubjobOk**.

4. Reliez le **tBonitaInstantiateProcess** au **tLogRow** à l'aide d'un lien**Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le composant **tBonitaDeploy** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **Bonita Runtime Jaas File**, spécifiez le chemin d'accès et le nom du fichier jaas.

   Dans le champ **Bonita Runtime Logging File**, spécifiez le chemin d'accès et le nom du fichier de log.

   Dans le champ **Business Archive**, spécifiez le chemin d'accès et le nom du processus Bonita.

3. Dans les champs **Username** et **Password**, saisissez vos informations d'authentification.

4. Double-cliquez sur le **tBonitaInstantiateProcess** pour ouvrir sa vue **Basic settings**.

5. Dans le champ **Bonita Runtime Jaas File**, spécifiez le chemin d'accès et le nom du fichier jaas.

   Dans le champ **Bonita Runtime Logging File**, spécifiez le chemin d'accès et le nom du fichier de log.

6. Dans les champs **Process Name** et **Process Version**, saisissez les informations de processus.

7. Dans les champs **Username** et **Password**, saisissez vos informations d'authentification.

8. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

9. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour obtenir un meilleur affichage.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** pour exécuter le Job.

```
**** Instance Bonita_V523_Process--1.0--6 created ****
.---------------------------.
|    show_instance_uuid     |
|=--------------------------=|
|ProcessInstanceUUID        |
|=--------------------------=|
|Bonita_V523_Process--1.0--6|
'---------------------------'

[statistics] disconnected
```

   Comme affiché ci-dessus, l'instance est créée et l'UUID est écrit en sortie.

tCentricCRMInput

Propriétés du tCentricCRMInput

Famille du composant	Business/CentricCRM	
Fonction	Le composant **tCentricCRMInput** se connecte à un module de la base de données CentricCRM via le service Web adéquat.	
Objectif	Le **tCentricCRMInput** permet d'extraire les données d'une base de données Centric CRM à l'aide d'une requête.	
Basic settings	*CentricCRM URL*	Saisissez l'URL du service Web permettant de se connecter à la base de données CentricCRM.
	Module	Sélectionnez le module adéquat dans la liste.
	Server	Saisissez l'adresse IP du serveur de base de données.
	UserID et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. In this component the schema is related to the Module selected.
	Query condition	Saisissez la requête permettant de sélectionner les données à extraire.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire.
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tCentricCRMOutput

Propriétés du tCentricCRMOutput

Famille du composant	Business/CentricCRM	
Fonction	Le composant **tCentricCRMOutput** écrit dans un module de la base de données CentricCRM via le service Web adéquat.	
Objectif	Le **tCentricCRMOutput** permet d'écrire des données dans une base de données CentricCRM.	
Basic settings	*CentricCRM URL*	Saisissez l'URL du service Web permettant de se connecter à la base de données CentricCRM.
	Module	Sélectionnez le module adéquat dans la liste.
	Server	Saisissez l'adresse IP du serveur de base de données.
	UserID et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Action	Vous pouvez effectuer l'une des opérations suivantes sur les données du module CentricCRM : **Insert** : insérer les données. **Update** : mettre les données à jour. **Delete** : supprimer les données.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tHL7Input

Propriétés du tHL7Input

Famille de composant	Business > Healthcare / Unstructured > HL7	
Fonction	Lit un fichier de structure HL7 et extrait des données ligne par ligne.	
Objectif	Ce composant lit un fichier HL7 structuré ou un flux de données ligne par ligne pour le scinder en champs et envoie les champs tels que définis dans le schéma au composant suivant du Job via une connexion **Row**.	
Basic settings	*Property Type*	Peut être Built-In ou Repository.
		Built-In : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant connexion de base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Multi Schemas Editor	L'éditeur **[Multi Schema Editor]** vous permet de construire et de configurer le flux de données dans un fichier délimité multi-structure, pour associer un schéma par sortie.
	Segment Lists	**Connection** : Les colonnes sont automatiquement récupérées à partir du fichier d'entrée. Le nom de la colonne est le nom du segment. **Column Mapping** : Dans ce tableau, le mapping est récupéré de celui que vous avez défini dans l'éditeur.
	Not Validate HL7 Message	Cochez cette case si vous ne souhaitez pas valider les messages HL7.
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Encoding	Sélectionnez le type d'encodage à partir de la liste ou choisissez l'option **Custom** pour le définir manuellement. Ce champ doit obligatoirement être renseigné.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Généralement utilisé en tant que composant d'entrée. Un composant de sortie est requis.

Scénario : Retrouver des informations concernant les patients et les événements à partir d'un fichier HL7

Ce scénario décrit un Job de quatre composants, dont le but est de retrouver des informations concernant des patients et des événements à partir d'un fichier HL7.

Configurer l'éditeur

1. A partir de la **Palette**, glissez-déposez un composant **tHL7Input**, et trois **tLogRow** dans l'espace de modélisation graphique.

2. Double-cliquez sur le composant **tHL7Input** afin d'ouvrir son éditeur.

3. Dans le champ **File path**, cliquez sur **[Browse...]**, parcourez votre répertoire, et sélectionnez votre fichier HL7.

4. Dans la zone **File Setting**, saisissez le caractère de début de votre segment (**Start character**), et le caractère de fin de votre segment (**End character**).

5. Dans la zone **Schema View**, dans la liste **Segment(As Schema)**, sélectionnez **MSH**.

6. Glissez-déposez les segments *MSH-3(1)-1-1[IS]* et *MSH-7(1)-1-1ST]* de la zone **Message View** dans la zone **Schema View** et modifiez la valeur de la colonne **User Column** à *MSH_LABEL* et *MSH_DATE* respectivement.

7. Dans la zone **Schema View**, dans la liste **Segment(As Schema)**, sélectionnez **EVN**.

8. Glissez-déposez les segments *EVN-1(1)-1-1[ID]* et *EVN-2(1)-1-1ST* de la zone **Message View** dans la zone **Schema View** et modifiez la valeur de la colonne **User Column** à *EVN_ID* et *EVN_DATE*, respectivement.

9. Dans la zone **Schema View**, dans la liste **Segment(As Schema)**, sélectionnez **PID**.

10. Glissez-déposez les segments suivants de la zone **Message View** dans la zone **Schema View** : *PID-1(1)-1-1[SI]*, *PID-5(1)-1-1ST*, *PID-5(1)-2-1ST*, *PID-5(1)-3-1ST*, *PID-5(1)-4-1ST*, *PID-5(1)-5-1ST*, *PID-7(1)-1-1ST*, *PID-8(1)-1-1[IS]*, puis modifiez la valeur de la colonne **User Column** à *PID_ID*, *PID_FNAME*, *PID_LNAME*, *PID_STNUMBER*, *PID_STREET*, *PID_CITY*, *PID_BDAY*, *PID_GENDER*, respectivement.

S'il est disponible, utilisez le bouton **Auto map!** situé en bas à gauche de la fenêtre, il permet d'effectuer l'opération de mapping automatiquement.

11. Cliquez sur **OK** pour fermer l'éditeur.

Configurer les composants de sortie

1. Double-cliquez sur le **tLogRow_1** pour afficher sa vue **Basic settings** et configurer ses propriétés.
2. Sélectionnez l'option **Table (print values in cells of a table)** afin d'afficher les résultats d'exécution dans un tableau.
3. Répétez l'opération pour le **tLogRow_2** et le **tLogRow_3**.

Exécution du Job

1. Reliez le **tHL7Input** aux trois **tLogRow**, en utilisant respectivement les liens **MSH**, **EVN** et **PID**.
2. Sauvegardez votre Job, et appuyez sur **F6** pour l'exécuter.

```
Starting job HL7 at 10:15 02/04/2010.

.---------+------------.
|      tLogRow_1       |
|=---------+-----------=|
|MSH_LABEL|MSH_DATE    |
|=---------+-----------=|
|MedSeries|200903230934|
'---------+------------'

.------+---------+---------+------------+-----------+---------+--------+----------.
|                                   tLogRow_2                                      |
|=-----+---------+---------+------------+-----------+---------+--------+---------=|
|PID_ID|PID_FNAME|PID_LNAME|PID_STNUMBER|PID_STREET |PID_CITY |PID_BDAY|PID_GENDER|
|=-----+---------+---------+------------+-----------+---------+--------+---------=|
|1     |Marie    |Charlotte|456         |main_street|Richemont|19770202|F         |
'------+---------+---------+------------+-----------+---------+--------+----------'
.------+-----------.
|     tLogRow_3     |
|=-----+-----------=|
|EVN_ID|EVN_DATE   |
|=-----+-----------=|
|A31   |200903230934|
'------+------------'
Job HL7 ended at 10:15 02/04/2010. [exit code=0]
```

La console affiche les trois tables des **tLogRow**, qui retournent différents types d'informations. La première montre l'en-tête du message et sa date. La seconde retourne les informations concernant le patient. La troisième affiche l'ID de l'événement ainsi que sa date.

tHL7Output

Propriétés du tHL7Output

Famille de composant	Business > Healthcare / Unstructured > HL7	
Fonction	Ecrit un fichier de structure HL7 et insère des données ligne par ligne.	
Objectif	Ce composant écrit un fichier HL7 structuré selon la norme HL7.	
Basic settings	*Property Type*	Peut être Built-In ou Repository.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema(s)	**Schema** : Définissez le nœud où stocker les données de la ligne parente. **Parent row** : Flux de données source.
	File Name/Output Stream	Parcourez le répertoire où stocker les fichiers générés.
	Configure HL7 Tree	Ouvre l'interface dédiée pour vous aider à mettre en place le mapping HL7.
	HL7 Version	Sélectionnez dans la liste votre version de HL7.
Advanced settings	*Create directory only if it does not exist*	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Encoding	Sélectionnez le type d'encodage à partir de la liste ou choisissez l'option **Custom** pour le définir manuellement. Ce champ doit obligatoirement être renseigné.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Généralement utilisé en tant que composant de sortie. Un composant d'entrée est requis.	
Limitation	Le nom de chaque lien d'entrée doit être le même que le nom du segment HL7 correspondant.	

Scénario : Générer un accusé de réception (Acknowledgment) HL7

Ce scénario, comprenant trois composants, permet de générer un accusé de réception HL7, ou Acknowledgment (ACK).

Construire le Job

1. Déposez deux composants **tFixedFlowInput** et un composant **tHL7Output** dans l'espace de modélisation graphique.
2. Renommez les deux **tFixedFlowInput** en **MSH** et **MSA** respectivement.
3. Reliez le **MSH** et le **MSA** au **tHL7Output** à l'aide d'un lien **Row** > **Main**.
4. Double-cliquez sur le lien entre le **MSH** et le **tHL7Output** et renommez-le *MSH*.
5. Double-cliquez sur le lien entre le **MSA** et le **tHL7Output** et renommez-le *MSA*.

Configurer les composants

1. Cliquez sur le **tHL7Output** pour ouvrir sa vue **Basic settings**.

Dans le champ **File Name/Output Stream**, spécifiez le répertoire de sortie de l'accusé de réception HL7.

Dans la liste déroulante **HL7 Version**, sélectionnez votre version de HL7, *2.3* dans cet exemple.

2. Cliquez sur le bouton **Configure HL7 Tree** pour ouvrir l'éditeur de l'arborescence HL7.

   Cliquez-droit sur le nœud **rootTag** dans le panneau **Linker Target** et sélectionnez **import HL7 Structure** dans le menu contextuel.

3. Dans la fenêtre **[Open]**, sélectionnez le fichier contenant l'accusé de réception (ACK) HL7 suivant.

```
MSH|^~\&|CATH|StJohn|AcmeHIS|StJohn|20130808||ACK|MSGID8|P|2.3
MSA|AA|MSGID8
```

   Pour plus d'informations concernant les messages ACK HL7, consultez http://www.hl7standards.com/blog/2007/02/01/ack-message-original-mode-acknowledgement/ (en anglais).

   Cliquez sur le bouton **Open** pour importer la structure de ce fichier.

Cliquez sur **Ok** pour fermer l'éditeur d'arborescence HL7. Deux lignes, **row_MSH_1** et **row_MSA_1** sont ajoutées à la table **Schema(s)** dans la vue **Basic settings** du **tHL7Output**.

4. Dans la table **Schema(s)**, cliquez dans la colonne **Parent row** pour la ligne **row_MSH_1** et sélectionnez **MSH** dans la liste déroulante.

   Répétez l'opération afin de sélectionner **MSA** dans la liste déroulante de la colonne **Parent row** pour la ligne **row_MSA_1**.

5. Dans la table **Schema(s)**, sélectionnez la cellule **row_MSH_1** et cliquez sur le bouton **[...]** de la cellule afin d'ouvrir l'éditeur de schéma.

Cliquez sur le bouton puis sur **OK** dans la boîte de dialogue afin de copier les colonnes du segment **MSH** du schéma de sortie dans le schéma d'entrée.

Répétez l'opération pour la cellule **row_MSA_1** et copiez les colonnes du segment **MSA** du schéma de sortie au schéma d'entrée.

Ainsi, vous n'avez pas besoin de définir manuellement les colonnes du schéma d'entrée, colonne par colonne.

6. Double-cliquez sur le composant **MSH** pour ouvrir sa vue **Basic settings**.

Sélectionnez l'option **Use Single Table** dans la zone **Mode** et renseignez la colonne **Value** pour chaque ligne:

- **MSH_1_1__1_1_ST_**: `"|"`
- **MSH_2_1__1_1_ST_**: `"^~\\&"`, assurez-vous d'utiliser le caractère d'échappement avant \, en mode **Use Single Table** ou **Use Inline Table**
- **MSH_9_1__1_1_ID_**: `"ACK"`
- **MSH_10_1__1_1_ST_**: `"MSGID66"`
- **MSH_11_1__1_1_ST_**: `"P"`
- **MSH_12_1__1_1_ID_**: `"2.3"`

Les colonnes ci-dessus doivent avoir des valeurs spécifiques pour le segment MSH. Pour plus d'informations, consultez http://www.hl7.org/documentcenter/public_temp_194525B1-1C23-BA17-0C3D37E51D9F8BA0/wg/conf/HL7MSH.htm (en anglais). Renseignez les valeurs pour les autres colonnes avec des guillemets doubles `""`.

7. Double-cliquez sur le composant **MSA** pour ouvrir sa vue **Basic settings**.

Sélectionnez l'option **Use Single Table** dans la zone **Mode** et renseignez la colonne **Value** pour chaque ligne :

- **MSA_1_1__1_1_ID_**: `"AA"`
- **MSA_2_1__1_1_ST_**: `"MSGID66"`

Les colonnes ci-dessus doivent avoir des valeurs spécifiques pour le segment MSA. Pour plus d'informations, consultez http://www.hl7standards.com/blog/2007/02/01/ack-message-original-mode-acknowledgement/ (en anglais).

Exécuter le Job

1. Appuyez sur **Ctrl + S** afin de sauvegarder le Job.
2. Appuyez sur **F6** afin d'exécuter le Job.

```
Starting job tHL7Output at 10:24 14/11/2013.

[statistics] connecting to socket on port 3344
[statistics] connected
[statistics] disconnected

Job tHL7Output ended at 10:24 14/11/2013. [exit
code=0]
```

Comme montré précédemment, le Job s'est correctement exécuté.

3. Ouvrez le message ACK HL7 généré :

```
hl7Out.txt
1  MSH|^~\&||||||ACK|MSGID66|P|2.3
2  MSA|AA|MSGID66
```

Comme montré précédemment, le message généré ne contient que les informations obligatoires et est totalement conforme aux standards HL7.

Scénario associé

Pour un scénario utilisant un autre composant HL7, consultez *Scénario : Retrouver des informations concernant les patients et les événements à partir d'un fichier HL7*.

tMarketoInput

Propriétés du tMarketoInput

Famille de composant	Business/Cloud	
Fonction	Le composant **tMarketoIntput** récupère des données à partir d'un serveur Web Marketo.	
Objectif	Ce composant vous permet de récupérer des données à partir d'une base de données Marketo sur un serveur Web.	
Basic settings	*Endpoint address*	URL du service Web Marketo appelé par l'API ou les API SOAP.
	Secret key	Saisissez le code d'authentification crypté fourni par Marketo. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour obtenir cette information, veuillez contacter le support Marketo via `<support@marketo.com>`.
	Client Access ID	Saisissez l'identifiant utilisateur permettant d'accéder au service Web Marketo. Pour obtenir cette information, veuillez contacter le support Marketo via `<support@marketo.com>`.
	Operation	Les options de cette liste vous permettent de synchroniser les données des prospects entre Marketo et un système externe. **getLead** : Cette option récupère les informations des prospects et de leur activité dans la base de données Marketo. **getMultipleLeads** : Cette opération récupère les enregistrements des prospects dans des lots. **getLeadActivity** : Cette opération récupère l'historique des enregistrements d'activité pour un prospect unique identifié par la clé fournie. **getLeadChanges** : Cette opération vérifie les changements des données de prospects dans la base de données Marketo.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au

		Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Columns Mapping	Vous pouvez configurer les conditions de mapping en modifiant le schéma dans **Edit Schema**. Par défaut, les noms des colonnes des champs contenus dans **Column** sont les mêmes que ceux du schéma. Puisque certains noms de colonnes de la base de données Marketo contiennent des blancs non acceptés par le schéma du composant, vous devez spécifier les champs de colonnes correspondants dans le champ **Columns in Marketo**. Si les noms de colonnes définis dans le schéma sont les mêmes que ceux de la base de données Marketo, comme par exemple "Email", alors il n'est pas nécessaire de configurer les conditions de mapping.
	Lead Selector	Sélectionnez dans la liste l'un des types de sélecteur de prospects : • **LeadKeySelector** : sélectionnez ce type afin de récupérer les enregistrements des prospects identifiés par la **LeadKey** spécifiée. • **StaticListSelector** : sélectionnez ce type afin de récupérer les enregistrements correspondant à la liste statique spécifiée. Lorsque cette option est sélectionnée, vous devez spécifier une liste statique en sélectionnant **List Name** ou **List ID** dans la liste affichée et saisir la valeur dans le champ à côté de la liste. • **LastUpdateAtSelector** : sélectionnez cet type afin de récupérer les enregistrements des prospects ayant été mis à jour à des dates spécifiques. Cette liste s'affiche uniquement lorsque l'option **getMultipleLeads** est sélectionnée dans la liste **Operation**.
	LeadKey type	Types de données de **LeadKey** supportés par la base de données Marketo. Cette liste s'affiche lorsque : • **getLead** ou **getLeadActivity** est sélectionné dans la liste **Operation**, ou • **getMultipleLeads** est sélectionné dans la liste **Operation** et **LeadKeySelector** est sélectionné dans la liste **Lead Selector**.
	LeadKey value	Valeur de **LeadKey**. Le champ est disponible uniquement lorsque **getLead** ou **getLeadActivity** est sélectionné dans la liste **Operation**.
	LeadKey values	Spécifiez les valeurs de **LeadKey** en sélectionnant une colonne du flux d'entrée contenant les valeurs **LeadKey**. Cette liste s'affiche uniquement lorsque l'option **getMultipleLeads** est sélectionnée dans la liste **Operation** et **LeadKeySelector** dans la liste **Lead Selector**.
	Set Include Types	Cochez cette case pour inclure les types de contenu de l'activité des prospects (LeadActivity) à récupérer. Cliquez sur le bouton **[+]** sous la zone **Include Types** pour ajouter des types en les sélectionnant dans la liste.

		Cette case est disponible uniquement si vous avez sélectionné **getLeadActivity** ou **getLeadChanges** dans la liste **Operation**.
	Set Exclude Types	Cochez cette case pour exclure les types de contenu de l'activité des prospects (**LeadActivity**) à récupérer. Cliquez sur le bouton **[+]** sous la zone **Exclude Types** pour ajouter des types en les sélectionnant dans la liste. Cette case est disponible uniquement si vous avez sélectionné **getLeadActivity** ou **getLeadChanges** dans la liste **Operation**.
	Oldest Updated At	Saisissez l'heure de la mise à jour la plus ancienne afin de récupérer uniquement les prospects mis à jour depuis le moment spécifié. Ce champ supporte le format de date et heure supporté par Marketo. Pour plus d'informations concernant le format de date supporté, consultez http://developers.marketo.com/documentation/soap/getmultipleleads/ (en anglais). Ce champ est disponible uniquement lorsque l'option **getMultipleLeads** est sélectionnée dans la liste **Operation** et **LastUpdateAtSelector** dans la liste **Lead Selector**.
	Last Updated At	Saisissez la date de la dernière mise à jour afin de ne récupérer que les prospects mis à jour avant le moment spécifié. Ce champ supporte le format de date et heure supporté par Marketo. Pour plus d'informations concernant le format de date supporté, consultez http://developers.marketo.com/documentation/soap/getmultipleleads/ (en anglais). Ce champ est disponible uniquement si vous avez sélectionné **getMultipleLeads** dans la liste **Operation** et **LastUpdateAtSelector** dans la liste **Lead Selector**.
	Oldest create date	Saisissez la date et l'heure de la première création afin de récupérer les données depuis le moment spécifié. Le format de date est YYYY-MM-DD HH:MM:SS Z. Ce champ est affiché uniquement lorsque vous sélectionnez **getLeadChanges** dans la liste **Operation**.
	Latest create date	Saisissez la date et l'heure de la dernière création afin de récupérer les données avant le moment spécifié. Le format de date est YYYY-MM-DD HH:MM:SS Z. Ce champ est affiché uniquement lorsque vous sélectionnez **getLeadChanges** dans la liste **Operation**. Les dates **Oldest create date** et **Latest create date** peuvent être spécifiés ensemble ou séparément.
	Batch Size	Limite maximale du nombre des données de prospects à récupérer par lot. Ce champ est disponible uniquement si : • vous avez sélectionné la case **getMultipleLeads** dans la liste **Operation** et **StaticListSelector** ou **LastUpdateAtSelector** dans la liste **Lead Selector**. • vous avez sélectionné la case **getLeadActivity** ou **getLeadChanges** dans la liste **Operation**.
	Timeout (milliseconds)	Saisissez en millisecondes le délai avant suspension de la requête sur le service Web Marketo. Le Job s'arrête lorsque l'erreur "Timeout exception" survient.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Au besoin, vous pouvez récupérer les lignes en erreur via un lien **Row** > **Reject**.

Advanced settings	*Max number of reconnection attempts*	Spécifiez le nombre maximum de fois que vous souhaitez vous reconnecter au serveur Web Marketo en cas de perte de connexion.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_CALL** : nombre d'appels. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de d'entrée. Un composant de sortie est nécessaire.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Insérer et récupérer des données d'une base de données Marketo via un système externe*.

tMarketoListOperation

Propriétés du tMarketoListOperation

Famille de composant	Business/Cloud	
Fonction	Le composant **tMarketoListOperation** ajoute/supprime un ou plusieurs prospect(s) dans une liste de la base de données Marketo. Il vérifie également si un ou plusieurs prospect(s) existe(nt) dans cette liste.	
Objectif	Ce composant vous permet d'ajouter ou de supprimer un ou plusieurs prospect(s) dans une liste de la base de données Marketo. Vous pouvez également vérifier l'existence d'un ou plusieurs prospect(s) dans la liste.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur **Edit schema** afin de visualiser le schéma.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Endpoint address	URL du service Web Marketo appelé par l'API ou les APIs SOAP.
	Secret key	Saisissez le code d'authentification crypté fourni par Marketo. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour obtenir cette information, veuillez contacter le support Marketo via `<support@marketo.com>`.
	Client Access ID	Saisissez l'identifiant utilisateur permettant d'accéder au service Web Marketo. Pour obtenir cette information, veuillez contacter le support Marketo via `<support@marketo.com>`.
	Operation	Les options de cette liste vous permettent d'effectuer un ajout/une suppression d'un ou plusieurs prospect(s) dans une liste d'une base de données Marketo. Vous pouvez également vérifier l'existence d'un ou plusieurs prospect(s) dans une liste de la base de données Marketo. **addTo** : Cette option ajoute un ou plusieurs prospect(s) dans une liste de la base de données Marketo. **isMemberOf** : Cette option recherche dans la base de données si le(s) prospect(s) spécifique(s) existe(nt) dans la base de données.

		removeFrom : cette opération supprime un ou plusieurs prospect(s) d'une liste dans la base de données Marketo.
	Add or remove multiple leads	Cochez cette case afin d'ajouter ou de supprimer plusieurs prospect(s) d'une liste dans la base de données Marketo. Cette case est disponible uniquement si vous avez sélectionné **addTo** ou **removeFrom** dans la liste **Operation**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Au besoin, vous pouvez récupérer les lignes en erreur via un lien **Row** > **Reject**.
	Timeout (milliseconds)	Saisissez en millisecondes le délai avant suspension de la requête sur le service Web Marketo. Le Job s'arrête lorsque l'erreur "Timeout exception" survient.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**NB_CALL** : nombre d'appels. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant intermédiaire. Un composant d'entrée et un composant de sortie sont nécessaires.	

Scénario : Ajouter un enregistrement de prospects dans une liste de la base de données Marketo

Le scénario suivant décrit un Job a trois composants qui ajoute un enregistrement de prospects dans une liste de la base de données Marketo.

Construire le Job

1. Déposez les composants **tMarketoListOperation**, **tFixedFlowInput** et **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Connectez le **tFixedFlowInput** au **tMarketoListOperation** à l'aide d'un lien **Row** > **Main**.

3. Connectez le **tMarketoListOperation** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer le composant d'entrée

1. Double-cliquez sur le composant **tFixedFlowInput** afin de définir ses propriétés dans sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir manuellement le schéma.

3. Cliquez sur le bouton **[+]** afin d'ajouter quatre colonnes : *ListKeyType*, *ListKeyValue*, *LeadKeyType* et *LeadKeyValue*. Laissez leurs paramètres par défaut. Cliquez sur **OK** pour sauvegarder la configuration.

4. Dans la zone **Mode**, sélectionnez **Use Inline Table**.

5. Cliquez sur le bouton **[+]** pour ajouter une ligne, puis remplissez-la avec les valeurs respectives. Dans cet exemple, ces valeurs sont : *MKTOLISTNAME* pour **ListKeyType**, *bchenTestList* pour **ListKeyValue**, *IDNUM* pour **LeadKeyType** et *308408* pour **LeadKeyValue**.

Configurer le tMarketoListOperation

1. Double-cliquez sur le composant **tMarketoListOperation** pour définir ses propriétés dans la vue **Basic settings**.

2. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma défini dans le **tFixedFlowInput**.

3. Saisissez *1* dans le champ **Number of rows**.

4. Dans le champ **Endpoint address**, saisissez l'URL du serveur Web Marketo. Dans cet exemple, saisissez *https://na-c.marketo.com/soap/mktows/1_5*.

   Notez que l'URL utilisée dans ce scénario a pour but la démonstration uniquement.

5. Dans le champ **Secret key**, saisissez le code d'authentification crypté fourni par Marketo. Dans cet exemple, le code est *46440763770355404 4DD11AA2211998*.

6. Saisissez l'ID de l'utilisateur dans le champ **Client Access ID**. Dans cet exemple, l'ID est *mktodemo41_785133934D1A219*.

7. Dans la liste **Operation**, sélectionnez **addTo**.

8. Saisissez la limite avant suspension de la requête dans le champ **Timeout**. Dans cet exemple, utilisez le délai par défaut : *60000*.

Exécuter le Job

1. Double-cliquez sur le composant **tLogRow** afin de configurer ses propriétés dans la vue **Basic settings**.

2. Cliquez sur le bouton **Sync columns** pour récupérer le schéma défini dans le composant **tMarketoListOperation**.

3. Dans la zone **Mode**, sélectionnez **Table**.

4. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job MarketoListOperation at 17:30 12/05/2011.

[statistics] connecting to socket on port 3497
[statistics] connected
.------------+-------------+-----------+------------+-------.
|                         tLogRow_1                          |
|=-----------+-------------+-----------+------------+------=|
|ListKeyType |ListKeyValue |LeadKeyType|LeadKeyValue|Success|
|=-----------+-------------+-----------+------------+------=|
|MKTOLISTNAME|bchenTestList|IDNUM      |308408      |true   |
'------------+-------------+-----------+------------+-------'

[statistics] disconnected
Job MarketoListOperation ended at 17:30 12/05/2011. [exit
code=0]
```

L'ajout d'un enregistrement de prospects dans une liste de la base de données Marketo est affiché dans la console de la vue **Run**.

tMarketoOutput

Propriétés du tMarketoOutput

Famille de composant	Business/Cloud	
Fonction	Le composant **tMarketoOutput** envoie des données vers un serveur Web Marketo.	
Objectif	Ce composant vous permet d'écrire des données dans une base de données Marketo sur un serveur Web.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Endpoint address	URL du service Web Marketo appelé par l'API ou les APIs SOAP.
	Secret key	Saisissez le code d'authentification crypté fourni par Marketo. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour obtenir cette information, veuillez contacter le support Marketo via `<support@marketo.com>`.
	Client Access ID	Saisissez l'identifiant utilisateur permettant d'accéder au service Web Marketo. Pour obtenir cette information, veuillez contacter le support Marketo via `<support@marketo.com>`.
	Operation	Les options de cette liste vous permettent de synchroniser les données des prospects entre Marketo et un système externe.

		syncLead : Cette option sollicite une opération d'insertion ou de mise à jour pour un enregistrement de prospect. **syncMultipleLeads** : Cette option sollicite une opération d'insertion ou de mise à jour pour plusieurs enregistrements de prospects regroupés par lots.
	Columns Mapping	Spécifiez les colonnes correspondantes dans Marketo pour les colonnes listées dans les champs **Column**. Par défaut, les noms de colonnes dans les champs **Column** sont les mêmes que dans le schéma. Vous pouvez modifier les colonnes de mapping en modifiant le schéma. Notez que les deux colonnes suivantes sont en lecture seule et que vous ne pouvez pas les mapper vers des colonnes Marketo : • **Status** : statut d'exécution de l'opération courante pour l'enregistrement de données. • **ERROR_MSG** : message d'erreur généré lorsqu'une erreur survient. Puisque certains noms de colonnes dans la base de données Marketo contiennent des blancs non acceptés par le schéma du composant, vous devez spécifier les champs de colonnes correspondants dans le champ **Columns in Marketo**. Vous n'avez pas besoin de mapper les colonnes de schéma ayant le même nom dans la base de données Marketo.
	De-duplicate lead record on email address	Cochez cette case afin de dédoublonner et de mettre à jour les enregistrements des prospects en utilisant l'adresse e-mail. Décochez cette case si vous souhaitez créer un autre enregistrement de prospect contenant la même adresse e-mail. Cette case est disponible uniquement si vous avez sélectionné **syncMultipleLeads** dans la liste **Operation**.
	Batch Size	Limite maximale du nombre d'enregistrements par lot à synchroniser avec les données de prospects. Ce champ est disponible uniquement si vous avez sélectionné **syncMultipleLeads** dans la liste **Operation**.
	Timeout (milliseconds)	Saisissez en millisecondes le délai avant suspension de la requête sur le service Web Marketo. Le Job s'arrête lorsque l'erreur "Timeout exception" survient.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Au besoin, vous pouvez récupérer les lignes en erreur via un lien **Row** > **Reject**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_CALL** : nombre d'appels. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en tant que composant de sortie. Un composant d'entrée est nécessaire.

Scénario : Insérer et récupérer des données d'une base de données Marketo via un système externe

Le scénario suivant décrit un Job à cinq composants qui a pour but d'insérer des enregistrements de prospects dans une base de données Marketo et d'écrire ces enregistrements dans un fichier local. Au terme de ce processus, le nombre d'appels d'API est affiché dans la console de la vue **Run**.

Configurer votre Job

1. A partir de la **Palette**, déposez les composants **tMarketoOutput**, **tMarketoInput**, **tFileInputDelimited**, **tFileOutputDelimited** et **tJava** dans l'espace de modélisation graphique.

2. Connectez le composant **tFileInputDelimited** au **tMarketoOutput** à l'aide d'un lien **Row** > **Main**.

3. Connectez le composant **tMarketoInput** au **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main**.

4. Connectez le composant **tFileInputDelimited** au **tMarketoInput** à l'aide d'un lien **Trigger** > **OnSubjectOk**.

5. Connectez le composant **tMarketoInput** au **tJava** à l'aide d'un lien **Trigger** > **OnSubjectOk**.

Configurer le tFileInputDelimited

1. Double-cliquez sur le composant **tFileInputDelimited** pour afficher l'onglet **Basic settings** de sa vue **Component** et définir ses propriétés.

2. Cliquez sur le bouton **[...]** à côté du champ **File name/Stream** et sélectionnez un fichier local pour l'insertion de données. Pour ce scénario, il s'agit de *D:/SendData.csv*.

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour configurer le schéma manuellement.

4. Cliquez sur le bouton **[+]** pour ajouter quatre colonnes : *Id*, *Email*, *ForeignSysPersonId* et *ForeignSysType*. Configurez le **Type** de *Id* à **Integer** et laissez les autres colonnes telles qu'elles sont. Cliquez ensuite sur **OK** pour enregistrer les paramètres.

5. Dans le champ **Header,** saisissez *1* et laissez les autres paramètres tels qu'ils sont.

Configurer le tMarketoOutput

1. Double-cliquez sur le composant **tMarketoOutput** pour afficher l'onglet **Basic settings** de sa vue **Component** et définir ses propriétés.

2. Cliquez sur le bouton **Sync columns** pour récupérer le schéma défini dans le composant **tFileInputDelimited**. Renseignez le champ **Endpoint address** avec l'adresse URL du serveur Web Marketo. Pour ce scénario, saisissez *https://na-c.marketo.com/soap/demo/demo1*.

   Notez que l'URL utilisée dans ce scénario est essentiellement destinée à la démonstration.

3. Renseignez le champ **Secret key** avec le code d'authentification crypté fourni par Marketo. Pour ce scénario, saisissez *1234567894DEMOONLY987654321*.

4. Renseignez le champ **Client Access ID** avec l'identifiant utilisateur. Pour ce scénario, saisissez *mktodemo1_1234567894DEMOONLY987654321*.

5. A partir de la liste **Operation**, sélectionnez **syncMultipleLeads**. Saisissez le délai (en millisecondes) avant suspension de la requête dans le champ **Timeout (milliseconds)**. Pour ce scénario, utilisez la valeur par défaut : *600000*.

Configurer le tMarketoInput

1. Double-cliquez sur le composant **tMarketoInput** afin d'afficher l'onglet **Basic settings** de sa vue **Component** et définir ses propriétés.

2. Dans la liste **Operation**, sélectionnez **getLead**.

3. Dans le tableau **Columns Mapping**, saisissez *test@talend.com* dans la colonne **Columns in Marketo** correspondant à la colonne **Email**.

   Notez que les données utilisées dans ce scénario sont essentiellement destinées à la démonstration.

4. Dans la liste **LeadKey type**, sélectionnez **EMAIL** et saisissez *test@talend.com* dans le champ **LeadKey value**.

5. Laissez les autres paramètres tels qu'ils sont dans le composant **tMarketoOutput**.

Configurer le tFileOutputDelimited

1. Double-cliquez sur le composant **tFileOutputDelimited** pour afficher l'onglet **Basic settings** de sa vue **Component** et définir ses propriétés.

2. Cliquez sur le bouton **[...]** à côté du champ **File name** pour synchroniser les données dans un fichier local. Dans ce scénario, le chemin d'accès au fichier est *D:/ReceiveData.csv*.

3. Cliquez sur le bouton **Sync columns** et laissez les autres paramètres tels qu'ils sont.

Utiliser des scripts Java pour compter les appels d'APIs

1. Double-cliquez sur le composant **tJava** pour ajouter du code dans l'onglet **Basic settings** de sa vue **Component**.

2. Dans le champ **Code**, saisissez le code suivant afin de compter le nombre d'appels d'API effectués pendant les opérations sur les données :

```
System.out.println(("The Number of API calls for inserting
data to Marketo DB is:"));
System.out.println((Integer)globalMap.get("tMarketoOutput_1_NB_CALL"));
System.out.println(("The Number of API calls for data synchronization
from Marketo DB is:"));
System.out.println((Integer)globalMap.get("tMarketoInput_1_NB_CALL"));
```

Exécuter le Job

1. Sauvegardez votre Job.

2. Appuyez sur **F6** pour l'exécuter.

```
Id;Email;ForeignSysPersonId;ForeignSysType
308434;;;
308436;;;
308435;;;
308431;;;
308432;;;
308433;;;
```

Les enregistrements des prospects insérés dans la base de données Marketo sont écrits dans le fichier *D:/ReceiveData.csv*.

```
Starting job io at 18:03 25/03/2011.

[statistics] connecting to socket on port 3829
[statistics] connected
The Number of API calls for inserting data to
Marketo DB is:
1
The Number of API calls for data synchronization
from Marketo DB is:
1
[statistics] disconnected
Job io ended at 18:03 25/03/2011. [exit code=0]
```

Le nombre total d'appels d'APIs durant les opérations sur les données est affiché dans la console de la vue **Run**.

tMicrosoftCrmInput

Propriétés du tMicrosoftCrmInput

Famille de composant	Business	
Fonction	Le composant **tMicrosoftCrmInput** récupère des données d'une entité de la base de données MicrosoftCRM.	
Objectif	Ce composant permet d'extraire les données d'une base MicrosoftCRM selon certaines conditions définies dans des colonnes spécifiques.	
Basic settings	*Authentication Type*	Sélectionnez le type d'authentification correspondant à des modèles spécifiques de déploiement de CRM. • **ON_PREMISE** : Le CRM est installé de manière personnalisée sur les serveurs des clients et la licence est achetée avec les infrastructures nécessaires. • **ONLINE** : Les clients accèdent à un CRM faisant partie d'un service de souscription basé Cloud, hébergé par Microsoft, dans leurs centres de données.
	Microsoft WebService URL	Saisissez l'URL du service Web d'un serveur Microsoft CRM sur site. Ce champ apparaît uniquement lorsque l'option **ON_PREMISE** est sélectionnée dans la liste **Authentication Type**. Dans ce cas, la version supportée de l'API Microsoft CRM est **v2007** et le champ peut être renseigné par l'URL comme suit : *http://ServerAddress/MSCrmServices/2007/CrmService.asmx*.
	API Version	Sélectionnez une version de l'API Microsoft CRM dans la liste déroulante. La version peut être **v2007** ou **v2011**. Ce champ apparaît uniquement lorsque l'option **ONLINE** est sélectionnée dans la liste déroulante **Authentication Type**.
	OrganizeName	Renseignez le nom de l'utilisateur ou de l'organisme, défini par un administrateur, voulant accéder à la base de données MicrosoftCRM.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Domain	Saisissez le nom de domaine du serveur sur lequel est hébergée la base MicrosoftCRM. Ce champ apparaît uniquement lorsque l'option **ON_PREMISE** est sélectionnée dans la liste **Authentication Type**.
	Host	Saisissez l'adresse IP du serveur de base de données Microsoft CRM. Ce champ apparaît uniquement lorsque l'option **ON_PREMISE** est sélectionnée dans la liste **Authentication Type**.
	Port	Saisissez le numéro du port d'écoute du serveur. Ce champ apparaît uniquement lorsque l'option **ON_PREMISE** est sélectionnée dans la liste **Authentication Type**.

	Discovery service endpoint	Spécifiez l'endpoint du service de découverte de CRM Online. Ce champ apparaît uniquement lorsque l'option **ONLINE** est sélectionnée dans la liste **Authentication Type**.
	Time out (second)	Spécifiez le nombre de secondes d'attente d'un message de réponse du serveur du CRM avant fermeture de la connexion.
	Entity	Sélectionnez l'entité adéquate à partir de la liste déroulante. Si nécessaire, sélectionnez **CustomEntity** dans la liste, puis spécifiez un nom d'entité dans le champ **Custom Entity Name** et définissez le schéma de l'entité personnalisée.
	Custom Entity Name	Spécifiez le nom de l'entité personnalisée. Ce champ apparaît uniquement lorsque l'option **CustomEntity** est sélectionnée dans la liste **Entity**.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Pour ce composant, le schéma correspond à l'entité sélectionnée.
	Logical operator used to combine conditions	Sélectionnez l'opérateur logique utilisé pour combiner les conditions définies dans la table **Conditions**. Deux opérateurs sont disponibles : • **And** : l'enregistrement est inclus dans l'ensemble des résultats lorsque toutes les conditions sont respectées. • **Or** : l'enregistrement est inclus dans l'ensemble des résultats lorsqu'une des conditions est respectée.
	Conditions	Cliquez sur le bouton **[+]** pour ajouter autant de conditions que nécessaire. Les paramètres suivants doivent être configurés pour chaque condition. • **Input column** : Cliquez dans la cellule et sélectionnez la colonne du schéma d'entrée à laquelle attribuer la condition. • **Operator** : Cliquez dans la cellule et sélectionnez l'opérateur utilisé pour comparer la valeur de la colonne d'entrée à la valeur définie dans la colonne **Value**. La valeur de ce paramètre peut être **Equal**, **NotEqual**, **GreaterThan**, **LessThan**, **GreaterEqual**, **LessEqual** et **Like**. • **Value** : Saisissez la valeur entre guillemets, si nécessaire.

Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire.	
Limitation	Ce composant supporte Microsoft CRM 4.0 (2007) pour le type d'authentification **On-Premise** et Microsoft CRM 4.0 (2007) et 2011 pour le type d'authentification **Online**.	

Scénario : Ecrire des données dans une base de données Microsoft CRM et attribuer des conditions aux colonnes pour extraire des lignes spécifiques

Le scénario suivant décrit un Job à quatre composants qui a pour but l'écriture des données d'un fichier délimité en entrée dans une entité personnalisée d'une base de données MicrosoftCRM. Le Job permet ensuite d'extraire des lignes spécifiques jusque dans un fichier de sortie, en utilisant les conditions définies sur certaines colonnes d'entrée.

Si vous voulez écrire dans une Entité Personnalisée d'une base de données MicrosoftCRM, assurez-vous de nommer les colonnes conformément aux règles de syntaxe Microsoft, à savoir "name_columnname" en minuscules.

Configurer le Job

1. A partir de la **Palette**, déposez les composants suivants dans l'éditeur graphique : **tFileInputDelimited**, **tFileOutputDelimited**, **tMicrosoftCrmInput**, et **tMicrosoftCrmOutput**.

2. Connectez le composant **tFileInputDelimited** au **tMicrosoftCrmOutput** à l'aide d'un lien de type **Row Main**.

3. Connectez le composant **tMicrosoftCrmIntput** au **tFileOutputDelimited** à l'aide d'un lien de type **Row Main**.

4. Connectez le composant **tFileInputDelimited** au **tMicrosoftCrmInput** à l'aide d'un lien de type **OnSubjobOk**.

Configurer le tFileInputDelimited

1. Double-cliquez sur le composant **tFileInputDelimited** pour afficher la vue **Basic settings** et configurer ses paramètres de base.

2. Sélectionnez le mode **Repository** à partir de la liste déroulante **Property Type** si vous avez stocké les paramètres du fichier d'entrée sous le nœud **Metadata** de l'arborescence **Repository**. Dans le cas contraire, sélectionnez le mode **Built-In** et remplissez manuellement les champs correspondants. Dans cet exemple, les propriétés sont définies en mode **Built-In**.

3. Cliquez sur le bouton **[...]** à côté du champ **File Name/Input Stream** et parcourez vos dossiers jusqu'au fichier délimité contenant les données d'entrée. Dans cet exemple, le fichier d'entrée comprend les colonnes suivantes : *new_id*, *new_status*, *new_firstname*, *new_email*, *new_city*, *new_initial* et *new_zipcode*.

```
new_id;new_status;new_firstname;new_email;new_city;new_initial;new_zipcode
1;married;Paul;pnewman@comp.com;New York;P.M;55677
2;single;Raul;pnewman@comp.com;New York;R.L;55677
3;single;Mary;mnewman@comp.com;Chicago;M.B;66898
4;married;John;Jnewman@comp.com;Chicago,J.M;66898
5;single;Martin;mnewman@comp.com;Sunnyvale,M.P;33662
6;married;Janet;jnewman@comp.com;Sunnyvale,J.P;33662
7;married;Harry;hnewman@comp.com;New York;H.M;55677
8;married;Jerry;jnewman@comp.com;New York;J.M;55677
9;married;Alice;anewman@comp.com;New York;A.M;55677
10;single;Jack;jnewman@comp.com;New York;J.M;55677
```

4. Dans le champ **Row separator**, définissez le séparateur de lignes permettant d'identifier les fins de ligne. De même, pour **Field separator**, définissez le séparateur de champs permettant de délimiter les différents champs pour chaque ligne.

5. Définissez, si nécessaire, le nombre de lignes d'en-tête (**Header**) et de pied de page (**Footer**) à ignorer, ainsi que le nombre maximum de lignes (**Limit**) que vous souhaitez traiter. Dans cet exemple, aucun de ces trois champs ne sont définis.

6. Cliquez sur **Edit schema** pour ouvrir une boîte de dialogue dans laquelle vous pouvez définir le schéma d'entrée que vous voulez écrire dans la base de données MicrosoftCRM.

7.

8. Cliquez sur **OK** pour fermer la boîte de dialogue.

Configurer le tMicrosoftCrmOutput

1. Double-cliquez sur le composant **tMicrosoftCrmOutput** pour afficher la vue **Basic settings** et configurer ses paramètres de base.

2. Dans le champ **Microsoft Web Service URL**, saisissez l'URL du serveur Web Microsoft. Renseignez ensuite le nom (**Username**) ainsi que le mot de passe (**Password**) correspondants.

3. Dans le champ **OrganizeName**, saisissez l'identifiant qui porte les droits d'accès à la base de données MicrosoftCRM.

4. Dans le champ **Domain**, renseignez le nom de domaine du serveur sur lequel la base de données MicrosoftCRM est hébergée. Saisissez ensuite l'adresse IP (**Host**) et le numéro de port d'écoute (**Port**) du serveur.

5. A partir de la liste déroulante **Action**, sélectionnez l'opération que vous voulez effectuer. Dans cet exemple, le but est d'insérer (**Insert**) des données dans une entité personnalisée sur la base MicrosoftCRM.

6. Dans le champ **Time out**, définissez la limite de temps (en secondes) après laquelle le Job s'arrête automatiquement.

7. Dans la liste **Entity**, sélectionnez l'une des options proposées. Dans cet exemple, c'est l'option *CustomEntity* qui est sélectionnée.

   Lorsque l'option *CustomEntity* est sélectionnée, le champ **Custom Entity Name** apparaît, dans lequel vous devez saisir le nom de l'entité personnalisée.

   Le **Schema** est alors défini de manière automatique, en fonction de l'entité sélectionnée. Cliquez si nécessaire sur le bouton **Edit schema** pour afficher une boîte de dialogue à partir de laquelle vous pouvez modifier ce schéma et retirer les colonnes dont vous n'avez pas besoin en sortie.

8. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

Configurer le tMicrosoftCrmInput

1. Double-cliquez sur le composant **tMicrosoftCrmInput** pour afficher la vue **Basic settings** et configurer ses paramètres de base.

2. Sélectionnez le mode **Repository** à partir de la liste déroulante **Property Type** si vous avez stocké les paramètres du fichier d'entrée sous le nœud **Metadata** de l'arborescence **Repository**. Dans le cas contraire, sélectionnez le mode **Built-In** et remplissez manuellement les champs correspondants. Dans cet exemple, les propriétés sont définies en mode **Built-In**.

3. Dans le champ **Microsoft Web Service URL**, saisissez l'URL du serveur Web Microsoft. Renseignez ensuite le nom (**Username**) ainsi que le mot de passe (**Password**) correspondants.

4. Dans le champ **OrganizeName**, saisissez l'identifiant qui porte les droits d'accès à la base de données MicrosoftCRM.

5. Dans le champ **Domain**, renseignez le nom de domaine du serveur sur lequel la base de données MicrosoftCRM est hébergée. Saisissez ensuite l'adresse IP (**Host**) et le numéro du port d'écoute (**Port**) du serveur.

6. Dans la liste **Entity**, sélectionnez l'une des options proposées. Dans cet exemple, c'est l'option *CustomEntity* qui est sélectionnée.

7. Le **Schema** est alors défini de manière automatique, en fonction de l'entité sélectionnée, mais vous pouvez le modifier si nécessaire. Dans cet exemple, pour accéder à l'entité personnalisée, il faut définir le schéma manuellement. Copiez le schéma à sept colonnes à partir du composant **tMicrosoftCrmOutput** et collez-le dans la boîte de dialogue du **tMicrosoftCrmInput**.

8. Cliquez sur le bouton **OK** pour fermer la boîte de dialogue. Un message pop-up vous proposera de propager les modifications : cliquez sur le bouton **Yes**.

9. Dans la vue **Basic settings**, sélectionnez l'opérateur logique **And** ou **Or** à utiliser pour combiner les conditions définies dans les colonnes d'entrée. Dans cet exemple, pour définir deux conditions sur deux colonnes d'entrée différentes, il faut utiliser l'opérateur logique **And**.

10. Dans la zone **Condition**, cliquez sur le bouton **[...]** pour ajouter autant de lignes que nécessaire, puis cliquez dans chaque ligne et sélectionnez, à partir de la liste déroulante **Input column**, la colonne sur laquelle appliquer la condition. Dans cet exemple, les deux colonnes *new-city* et *new_id* portent des conditions. Le but est d'extraire toutes les lignes concernant les clients dont la ville correspond à "New York" et dont l'identifiant est supérieur à 2.

11. Cliquez dans chaque ligne et sélectionnez, à partir de la liste déroulante **Operator**, l'opérateur à utiliser pour relier la colonne d'entrée avec sa valeur associée. Dans cet exemple, l'opérateur **Equal** est sélectionné pour la colonne *new_city* et **Greater Than** pour *new_id*.

12. Cliquez dans chaque ligne et définissez, dans la colonne **Value**, la valeur des colonnes d'entrée : `New York` pour la colonne *new_city* et `2` pour *new_id* dans cet exemple. Il est possible d'utiliser une valeur fixe aussi bien qu'une valeur de contexte dans ce champ.

Configurer le tFileOutputDelimited

1. Double-cliquez sur le composant **tFileOutputDelimited** pour afficher la vue **Basic settings** et configurer ses paramètres de base.

2. Sélectionnez le mode **Built-In** à partir de la liste déroulante **Property Type** puis cliquez sur le bouton **[...]** à côté du champ **File Name** pour parcourir vos dossiers jusqu'au fichier de sortie.
3. Définissez les séparateurs de lignes (**Row separator**) et de champs (**Field separator**).
4. Cochez la case **Append** pour ajouter des lignes à la fin des entrées de la base de données.
5. Cochez la case **Include Header** si le fichier de sortie comprend un en-tête.
6. Cliquez sur le bouton **Sync columns** pour récupérer le schéma à partir du composant précédent.

Exécuter votre Job

- Enregistrez votre Job puis cliquez sur **F6** pour l'exécuter.

	A	B	C	D	E
1	7;married;Harry;hnewman@comp.com;New York;H.M;55677				
2	8;married;Jerry;jnewman@comp.com;New York;J.M;55677				
3	9;married;Alice;anewman@comp.com;New York;A.M;55677				
4	10;single;Jack;jnewman@comp.com;New York;J.M;55677				
5					
6					

Seuls les clients habitant la ville de New York, dont l'identifiant est supérieur à 2, apparaissent dans la liste du fichier de sortie stocké en local.

tMicrosoftCrmOutput

Propriétés du tMicrosoftCrmOutput

Famille de composant	Business	
Fonction	Le composant **tMicrosoftOutput** écrit des données dans une base de données MicrosoftCRM.	
Objectif	Ce composant permet d'écrire des données dans une base de données MicrosoftCRM.	
Basic settings	*Authentication Type*	Sélectionnez le type d'authentification correspondant à des modèles spécifiques de déploiement de CRM. • **ON_PREMISE** : Le CRM est installé de manière personnalisée sur les serveurs des clients et la licence est achetée avec les infrastructures nécessaires. • **ONLINE** : Les clients accèdent à un CRM faisant partie d'un service de souscription basé Cloud, hébergé par Microsoft, dans leurs centres de données.
	Microsoft WebService URL	Saisissez l'URL du service Web d'un serveur Microsoft CRM sur site. Ce champ apparaît uniquement lorsque l'option **ON_PREMISE** est sélectionnée dans la liste **Authentication Type**. Dans ce cas, la version supportée de l'API Microsoft CRM est **v2007** et le champ peut être renseigné par l'URL comme suit : *http://ServerAddress/MSCrmServices/2007/CrmService.asmx*.
	API Version	Sélectionnez une version de l'API Microsoft CRM dans la liste déroulante. La version peut être **v2007** ou **v2011**. Ce champ apparaît uniquement lorsque l'option **ONLINE** est sélectionnée dans la liste déroulante **Authentication Type**.
	OrganizeName	Renseignez le nom de l'organisme qui veut accéder à la base de données MicrosoftCRM.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Domain	Saisissez le nom de domaine du serveur sur lequel est hébergée la base MicrosoftCRM. Ce champ apparaît uniquement lorsque l'option **ON_PREMISE** est sélectionnée dans la liste **Authentication Type**.
	Host	Saisissez l'adresse IP du serveur de base de données Microsoft CRM. Ce champ apparaît uniquement lorsque l'option **ON_PREMISE** est sélectionnée dans la liste **Authentication Type**.
	Port	Saisissez le numéro du port d'écoute du serveur. Ce champ apparaît uniquement lorsque l'option **ON_PREMISE** est sélectionnée dans la liste **Authentication Type**.
	Discovery service endpoint	Spécifiez l'endpoint du service de découverte de CRM Online.

		Ce champ apparaît uniquement lorsque l'option **ONLINE** est sélectionnée dans la liste **Authentication Type**.
	Action	Vous pouvez effectuer l'une des opérations suivantes sur les données MicrosoftCRM : **Insert** : insérer les données. **Update** : mettre les données à jour. **Delete** : supprimer les données.
	Time out (second)	Spécifiez le nombre de secondes d'attente d'un message de réponse du serveur du CRM avant fermeture de la connexion.
	Entity	Sélectionnez l'entité adéquate à partir de la liste déroulante. Si nécessaire, sélectionnez **CustomEntity** dans la liste, puis spécifiez un nom d'entité dans le champ **Custom Entity Name** et définissez le schéma de l'entité personnalisée.
	Custom Entity Name	Spécifiez le nom de l'entité personnalisée. Ce champ apparaît uniquement lorsque l'option **CustomEntity** est sélectionnée dans la liste **Entity**.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Pour ce composant, le schéma correspond à l'entité sélectionnée.
	Lookup Type Mapping	Cliquez sur le bouton **[+]** pour ajouter autant de lignes que nécessaire afin d'établir le mapping entre les tables source et cible. Sélectionnez un objet de référence dans la liste déroulante **Input column** et saisissez le mot-clé des tables source dans le champ **Type**.
Advanced settings	*Reuse Http Client*	Cochez cette case afin de conserver la connexion courante ou décochez la case afin de libérer la connexion.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.
Limitation	Ce composant supporte Microsoft CRM 4.0 (2007) pour le type d'authentification **On-Premise** et Microsoft CRM 4.0 (2007) et 2011 pour le type d'authentification **Online**.

Scénario associé

Pour un scénario associé, consultez le composant **tMicrosoftCrmInput**.

tMSAXInput

Propriétés du tMSAXInput

Famille de composant	Business/ Microsoft AX	
Fonction	Le composant **tMSAXInput** se connecte au serveur MicrosoftAX.	
Objectif	Ce composant permet de lire les données d'un serveur MicrosoftAX et d'effectuer des requêtes sur ces données.	
Basic settings	*Property type*	Peut être Built-in ou Repository.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Version	Sélectionnez le type de connecteur métier à utiliser. Ce type peut être : • **Dynamics AX 4.0** • **Dynamics AX 2012**
Type **Dynamics AX 2012** uniquement.	*.NET Business Connector Assembly Path*	Parcourez votre système ou saisissez le chemin d'accès au fichier Assembly de votre connecteur métier .NET.
	Host	Saisissez l'adresse IP du serveur de MicrosoftAX. Lorsque vous utilisez le connecteur métier .NET, le Job correspondant doit être exécuté sur le serveur où est hébergé votre serveur Dynamics AX. Si l'édition de votre *Studio Talend* vous permet d'utiliser un JobServer pour exécuter ce Job, vous devez déployer ce JobServer sur le serveur de l'hôte de votre serveur Dynamics AX.
Type **Dynamics AX 2012** uniquement.	*Port*	Saisissez le numéro du port du connecteur .NET à utiliser.
Type **Dynamics AX 2012** uniquement.	*AOS Server Instance*	Saisissez le nom de l'ordinateur exécutant l'instance de l'Application Object Server (AOS) à laquelle vous devez vous connecter.
	Domain	Saisissez le nom de domaine sur lequel est hébergé le serveur MicrosoftAX.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Type **Dynamics AX 2012** uniquement.	*Company*	Saisissez le nom de votre entreprise.
Type **Dynamics AX 2012** uniquement.	*Language*	Saisissez la langue que vous souhaitez utiliser.
Type **Dynamics AX 2012** uniquement.	*Configuration File*	Spécifiez l'emplacement du fichier fournissant les paramètres à utiliser.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.

		Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à lire.
	Query	Saisissez votre requête SQL en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tMSAXOutput

Propriétés du tMSAXOutput

Famille de composant	Business/ Microsoft AX	
Fonction	Le composant **tMSAXOutput** se connecte au serveur MicrosoftAX.	
Objectif	Ce composant permet d'écrire les données d'un serveur MicrosoftAX.	
Basic settings	*Property type*	Peut être Built-in ou Repository.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Version	Sélectionnez le type de connecteur métier à utiliser. Ce type peut être : • **Dynamics AX 4.0** • **Dynamics AX 2012**
Type **Dynamics AX 2012** uniquement.	*.NET Business Connector Assembly Path*	Parcourez votre système ou saisissez le chemin d'accès au fichier Assembly de votre connecteur métier .NET.
	Host	Saisissez l'adresse IP du serveur de MicrosoftAX. Lorsque vous utilisez le connecteur métier .NET, le Job correspondant doit être exécuté sur le serveur où est hébergé votre serveur Dynamics AX. Si l'édition de votre *Studio Talend* vous permet d'utiliser un JobServer pour exécuter ce Job, vous devez déployer ce JobServer sur le serveur de l'hôte de votre serveur Dynamics AX.
Type **Dynamics AX 2012** uniquement.	*Port*	Saisissez le numéro du port du connecteur .NET à utiliser.
Type **Dynamics AX 2012** uniquement.	*AOS Server Instance*	Saisissez le nom de l'ordinateur exécutant l'instance de l'Application Object Server (AOS) à laquelle vous devez vous connecter.
	Domain	Saisissez le nom de domaine sur lequel est hébergé le serveur MicrosoftAX.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Type **Dynamics AX 2012** uniquement.	*Company*	Saisissez le nom de votre entreprise.
Type **Dynamics AX 2012** uniquement.	*Language*	Saisissez la langue que vous souhaitez utiliser.
Type **Dynamics AX 2012** uniquement.	*Configuration File*	Spécifiez l'emplacement du fichier fournissant les paramètres à utiliser.
	Table Name	Nom de la table à écrire.
	Action on data	Vous pouvez effectuer n'importe quelle opération sur les données d'un serveur MicrosoftAX. **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.

		Update : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations Update et Delete. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (**Advanced settings**) pour une utilisation optimisée de ces opérations.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
	Additional columns	Cette option vous permet d'utiliser des expressions locales afin d'effectuer des actions sur les colonnes. Par exemple, vous pouvez modifier les valeurs dans la table sélectionnée : Quand vous mettez à jour ou que vous effacez des données dans une colonne, cette option vous fournit d'autres possibilités concernant les clauses WHERE en utilisant différents opérateurs de la colonne **Operator**. **Name** : nom de la colonne du schéma devant être modifiée ou insérée comme nouvelle colonne. **Operator** : sélectionnez dans la liste l'opérateur que vous souhaitez utiliser avec la clause WHERE. Cette colonne n'est pas disponible quand vous utilisez **Insert** comme action sur les données. **Data type** : type de données. **Local expression** : type d'expression locale qui sera exécuté afin de modifier ou insérer la colonne de données correspondante, par

		exemple *row1.[row name]*. Ou bien appuyez sur **Ctrl+Espace** et sélectionnez une variable de contexte dans la liste. **Position** : sélectionnez dans la liste **Before**, **After** ou **Replace**, suivant l'action que vous voulez exécuter dans la colonne de référence. **Reference column** : type, dans une colonne de référence, que le composant peut utiliser pour insérer/remplacer la nouvelle colonne ou la colonne modifiée.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de sortie. Un composant de début est nécessaire.	

Scénario : Insérer des données dans une table définie sur le serveur Microsoft AX

Avant de pouvoir utiliser ce composant, vérifiez que vous avez installé et lancé le serveur Microsoft AX correctement.

Ce scénario décrit un Job à deux composants qui utilise le composant **tMSAXOutput** afin d'insérer une ligne dans une table définie stockée sur un serveur Microsoft AX et modifie ensuite les valeurs dans l'une des colonnes insérée.

Configurer le Job

1. Glissez le composant **tFixedFlowInput** ainsi que le **tMSAXOutput** de la **Palette** dans l'espace de modélisation.
2. Connectez les composants à l'aide d'un lien **Row** de type **Main**.

Configurer le tFixedFlowInput

1. Double-cliquez sur le **tFixedFlowInput** afin d'afficher la vue **Component** et de définir ses propriétés.

2. Sélectionnez le mode **Built-In** dans le champ **Schema** et cliquez sur le bouton **[...]** du champ **Edit schema** afin d'afficher une boîte de dialogue qui vous permettra de définir le schéma d'entrée.
3. Cliquez sur le bouton **[+]** vert pour ajouter des colonnes dans le schéma d'entrée, trois dans cet exemple, *name*, *city* et *street* (nom, ville et rue).
4. Cliquez sur **OK** pour fermer la boîte de dialogue. Une nouvelle boîte de dialogue s'ouvre et vous demande si vous voulez propager les modifications, cliquez sur **Yes** (Oui). Les colonnes du schéma s'affichent automatiquement dans le tableau **Values**.
5. Cliquez sur la colonne **Value** et saisissez une valeur pour chaque colonne d'entrée.

Configurer le tMSAXOutput

1. Double-cliquez sur le **tMSAXOutput** pour ouvrir la vue **Component** et définir ses propriétés.

2. Sélectionnez le mode **Built-In** dans le champ **Property Type**.

3. Dans le champ **Host**, saisissez l'adresse IP du serveur MicrosoftAX et, dans le champ **Domain**, saisissez le nom du domaine qui héberge le serveur MicrosoftAX.

4. Entrez votre nom d'utilisateur et votre mot de passe de connexion au serveur dans les champs correspondants, puis, dans le champ **Table Name**, saisissez le nom de la table dans la quelle vous souhaitez écrire des données.

5. Dans la liste **Action on data**, sélectionnez l'action que vous désirez exécuter, **Insert** dans cet exemple.

6. Cliquez sur **Sync columns** pour retrouver le schéma du composant précédent.

   Dans cet exemple, le but est de retrouver les trois colonnes d'entrée : *name*, *city* et *street* (nom, ville et rue) et d'écrire les données comprises dans les trois colonnes d'entrée du serveur MicrosoftAX sans effectuer de modification.

   Si nécessaire, cliquez sur le bouton **[...]** du champ **Edit schema** afin d'ouvrir une boîte de dialogue qui vous permettra de vérifier le schéma retourné.

7. Dans le tableau **Additional columns**, cliquez sur le bouton **[+]** afin d'ajouter une ligne dans laquelle vous pouvez définir les paramètres de la nouvelle colonne à ajouter à la ligne que vous souhaitez écrire dans la table *ADDRESS*.

8. Définissez un nom, un type de données, une position et une colonne de référence dans les colonnes correspondantes de la ligne ajoutée.

   Dans cet exemple, ajoutez une nouvelle colonne nommée "*address*" après la colonne "*street*".

9. Cliquez sur la colonne **Local expression** et appuyez sur **Ctrl+Espace** sur votre clavier pour ouvrir la liste des variables de contexte et sélectionnez : `StringHandling.UPCASE(row2.city)+"-"+row2.street`. Cette expression écrira le nom de la ville avec la première lettre en majuscule, suivi du nom de la rue, pour former l'adresse du *Bryant park*. La colonne *address* de cet exemple contiendra donc la chaîne de caractères suivante : *New York-Midtown Manhattan*.

Exécuter le Job

- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

  Le **tMSAXOutput** insère dans la table *ADDRESS* du serveur MicrosoftAX une ligne contenant les trois colonnes d'entrée *name*, *city* et *street*, en plus de la nouvelle colonne *address* qui contient à la fois le nom de la ville et le nom de la rue.

Scénario 2 : Effacer des données d'une table précise sur le serveur MicrosoftAX.

Avant de pouvoir utiliser ce composant, vérifiez que vous avez installé et lancé correctement le serveur MicrosoftAX.

Ce scénario décrit un Job à deux composants qui utilise le **tMSAXOutput** dans le but d'effacer toutes les lignes d'une table définie qui ne correspondent pas aux données contenues dans la colonne clé sur un serveur MicrosoftAX.

Dans cet exemple, le schéma d'entrée que vous utilisez est une colonne *address* (adresse) qui contient les données suivantes : *New York-Midtown Manhattan*. Vous allez effacer du serveur MicrosoftAX toutes les adresses qui ne sont pas identiques à celle-ci.

Configurer le Job

1. Glissez les composants **tFixedFlowInput** et **tMSAXOutput** de la **Palette** dans l'espace de modélisation graphique.

2. Connectez les deux composants à l'aide d'un lien **Row** de type **Main**.

Configurer le tFixedFlowInput

1. Double-cliquez sur le **tFixedFlowInput** pour afficher sa vue **Component** et définir ses propriétés.

2. Sélectionnez le mode **Built-In** dans le champ **Schema** et cliquez sur le bouton **[...]** du champ **Edit schema** pour ouvrir une boîte de dialogue où vous pourrez définir le schéma.

3. Cliquez sur le bouton **[+]** et ajoutez les colonnes d'entrée du schéma, *address* dans cet exemple.

4. Cliquez sur **OK** pour fermer la boîte de dialogue. Les colonnes du schéma s'affichent automatiquement dans la liste **Values**.

5. Cliquez sur la colonne **Value** et saisissez une valeur pour la colonne d'entrée.

Configurer la connexion au serveur MicrosoftAX

1. Double-cliquez sur le **tMSAXOutput** pour ouvrir sa vue **Component** et définir ses propriétés.

2. Sélectionnez le mode **Built-In** dans le champ **Property Type**.

3. Dans le champ **Host**, saisissez l'adresse IP du serveur MicrosoftAX.

4. Dans le champ **Domain**, entrez le nom du domaine qui héberge le serveur MicrosoftAX.

5. Saisissez votre nom d'utilisateur et votre mot de passe de connexion au serveur dans les champs correspondants.

6. Dans le champ **Table Name,** saisissez le nom de la table dans laquelle vous voulez supprimer des données, *ADDRESS* dans cet exemple.

Définir l'action sur les données

1. Dans la liste **Action on data**, sélectionnez l'action que vos souhaitez exécuter, **Delete** dans cet exemple.

2. Cliquez sur **Sync columns** pour retourner le schéma du composant précédent. Dans cet exemple, l'objectif est de récupérer la colonne d'entrée *address*.

3. Cliquez sur le bouton **[...]** du champ **Edit Schema** pour ouvrir la boîte de dialogue dans laquelle vous pouvez vérifier le schéma récupéré.

4. Dans le schéma de sortie, cochez la case **Key** (clé) à côté du nom de la colonne puis cliquez sur **OK** pour valider les changements et fermer la boîte de dialogue.

   Lorsque vous sélectionnez l'action sur les données **Delete**, vous devez définir la colonne de référence **Reference column** comme colonne clé afin que le **tMSAXOutput** supprime les lignes à partir de cette colonne clé.

5. Dans le tableau **Additional columns**, cliquez sur le bouton **[+]** pour ajouter une ligne et définir les paramètres que le composant utilisera comme base pour l'opération d'effacement.

6. Saisissez un nom (**name**), un opérateur (**operator**), un type de données (**data type**), une expression locale (**local expression**), une position (**position**) et une colonne de référence (**reference column**) dans les colonnes correspondantes à la ligne que vous venez d'ajouter.

   Dans cet exemple, l'objectif est d'effacer de la table *ADDRESS* du serveur MicrosoftAX toutes les lignes dans lesquelles la colonne *Address* n'est pas identique à l'adresse dans la colonne clé *address*, `New York-Midtown Manhattan`.

   Quand vous sélectionnez l'option **Delete**, vous devez toujours paramétrer **Position** à **Replace**. Sinon, les paramètres du tableau **Additional columns** ne seront pas pris en compte lors de l'exécution de votre Job.

Exécuter le Job

- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

  Le **tMSAXOutput** supprime de la table *ADDRESS* du serveur MicrosoftAX toutes les lignes pour lesquelles les données de la colonne *address* ne sont pas identiques à celles de la colonne clé.

tNetsuiteInput

Propriétés du tNetsuiteInput

Famille de composant	Business/NetSuite	
Fonction	Le composant **tNetsuiteInput** invoque le service SOAP NetSuite et récupère les données selon les conditions spécifiées.	
Objectif	Ce composant vous permet d'accéder au serveur SOAP NetSuite et d'en récupérer des données.	
Basic settings	*Username* et *Password*	Saisissez vos informations d'authentification pour accéder au service SOAP NetSuite. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Role	Saisissez l'ID de votre rôle NetSuite.
	Account	Saisissez votre nom de compte NetSuite.
	Record Type	Sélectionnez un type d'enregistrement à récupérer.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. *Le schéma de ce composant est pré-renseigné avec des colonnes correspondant aux champs des services NetSuite, qui varient selon le type d'enregistrement (**Record Type**) sélectionné. Vous pouvez supprimer les colonnes ne concernant pas votre opération, mais ne renommez pas de colonne du schéma, cela pourrait causer des erreurs d'exécution.*
	Search Criteria	Définissez les conditions de recherche afin de faire correspondre les champs pour récupérer les données. Cliquez sur le bouton **[+]** pour ajouter autant de lignes que nécessaire, chaque ligne pour une condition et définir chacune de vos conditions : • **Field Name** : sélectionnez un champ dans lequel rechercher selon la condition spécifiée. • **Operator** : sélectionnez un opérateur pour la condition de recherche.

		• **Value** : saisissez la valeur pour la condition de recherche. • **Value2** : saisissez la seconde valeur de la condition de la recherche, si approprié, selon l'opérateur sélectionné.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début dans un flux **Talend** et nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tNetsuiteOutput

Propriétés du tNetsuiteOutput

Famille de composant	Business/NetSuite	
Fonction	Le composant **tNetsuiteOutput** invoque le service SOAP Netsuite et insère, met à jour ou supprime des données sur le serveur SOAP NetSuite.	
Objectif	Ce composant vous permet d'accéder au serveur SOAP NetSuite et d'exécuter l'action définie sur les données, sur le serveur.	
Basic settings	*Username* et *Password*	Saisissez vos informations d'authentification pour accéder au service SOAP NetSuite. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Role	Saisissez l'ID de votre rôle NetSuite.
	Account	Saisissez votre nom de compte NetSuite.
	Action	Sélectionnez une action à effectuer dans la liste. • **Insert** : ajouter un nouvel enregistrement à vos données. Si un doublon est trouvé et que la case **Die on error** est cochée, le Job s'arrête. Si la case **Die on error** est décochée, le doublon est ignoré et le Job continue. • **Update** : apporter des modifications à un enregistrement existant. Si l'enregistrement correspondant à l'ID interne (Internal ID) spécifié n'existe pas et que la case **Die on error** est cochée, le Job s'arrête. Pour utiliser cette option, la colonne *InternalId* doit exister dans le schéma et être définie comme Key. • **Delete** : supprimer un enregistrement existant. Si l'enregistrement correspondant à l'ID interne (Internal ID) spécifié n'existe pas et que la case **Die on error** est cochée, le Job s'arrête. Pour utiliser cette option, la colonne *InternalId* doit exister dans le schéma et être définie comme Key. • **Upsert** : mettre à jour l'enregistrement correspondant à l'ID interne spécifié ou insérer un nouvel enregistrement s'il n'existe pas. Pour utiliser cette option, la colonne *InternalId* doit exister dans le schéma et être définie comme Key.
	Record Type	Sélectionnez le type d'enregistrement sur lequel travailler.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

		Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. *Le schéma de ce composant est pré-renseigné avec des colonnes correspondant aux champs des services NetSuite, qui varient selon le type d'enregistrement (**Record Type**) sélectionné. Vous pouvez supprimer les colonnes ne concernant pas votre opération, mais ne renommez pas de colonne du schéma, cela pourrait causer des erreurs d'exécution.*
	Die on error	Cette case est cochée par défaut. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **CURRENT_INTERNALID**: The internal ID of the record currently inserted or processed. This is a Flow variable and it returns an integer. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de fin dans un flux **Talend** et nécessite un composant d'entrée.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tOpenbravoERPInput

Propriétés du tOpenbravoERPInput

Famille de composant	Business	
Fonction	Le composant **tOpenbravoERPInput** se connecte à une entité de la base de données OpenbravoERP via le service Web adéquat.	
Objectif	Ce composant permet d'extraire les données d'une base OpenBravoERP selon certaines conditions définies dans des colonnes spécifiques.	
Basic settings	*Openbravo REST WebService URL*	Saisissez l'URL du service Web permettant de se connecter à la base de données OpenbravoERP.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Entity	Sélectionnez l'entité adéquate à partir de la liste déroulante.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Pour ce composant, le schéma correspond à l'entité sélectionnée.
	WHERE Clause	Saisissez votre clause WHERE.
	Order by	Cochez cette case afin de sélectionner comment ordonner les résultats (les éléments de la liste déroulante dépendent de l'entité sélectionnée). **Sort** : Dans cette liste, vous pouvez choisir de trier vos résultats de manière ascendante (**Ascending**) ou descendante (**Descending**).
	First result	Saisissez le numéro de la ligne que vous souhaitez récupérer en premier.

	Max result	Saisissez le nombre maximal de résultats que vous souhaitez récupérer.
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire.	

Scénario associé

Pour un scénario associé, consultez le composant **tMicrosoftCrmInput**.

tOpenbravoERPOutput

Propriétés du tOpenbravoERPOutput

Famille de composant	Business	
Fonction	Le composant **tOpenbravoERPOutput** écrit dans un objet de la base de données OpenbravoERP via le service Web adéquat.	
Objectif	Ce composant permet d'écrire des données dans une base de données OpenbravoERP.	
Basic settings	*Openbravo REST Webservice URL*	Saisissez l'URL du service Web permettant de se connecter à la base de données OpenbravoERP.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Action on data	Sélectionnez dans la liste si vous souhaitez mettre à jour ou créer des données (**Update/Create**), ou si vous préférez en supprimer (**Remove**).
	Use existing data file	Cochez cette case afin de sélectionner votre fichier en parcourant votre répertoire.
	Entity	Sélectionnez l'entité adéquate à partir de la liste déroulante.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier.	

	NB_LINE_CREATED : nombre de lignes créées. Cette variable est une variable **After** et retourne un nombre entier. **NB_LINE_REMOVED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un nombre entier. **NB_LINE_UNAUTHORIZED** : nombre de lignes non autorisées. Cette variable est une variable **After** et retourne un nombre entier. **NB_LINE_FAILED** : nombre de lignes en erreur. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.

Scénario associé

Pour un scénario associé, consultez le composant **tMicrosoftCrmInput**.

tSageX3Input

Propriétés du tSageX3Input

Famille de composant	Business/Sage X3	
Fonction	Le composant **tSageX3Input** tire profit du Service Web fourni par un serveur Web Sage X3 pour extraire des données du système Sage X3 (serveur X3).	
Objectif	Le composant **tSageX3Input** extrait des données d'un système Sage X3 donné.	
Basic settings	*Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Endpoint address	Saisissez l'adresse du Service Web fourni par le serveur Web Sage X3 donné.
	Username et *Password*	Saisissez les informations d'authentification pour le Service Web, définies lors de la configuration du Serveur Web Sage X3. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Language	Saisissez le nom du code langue X3 utilisé pour le démarrage du groupe de connexions.
	Pool alias	Saisissez le nom du pool de connexion distribuant les requêtes reçues aux connexions disponibles. Ce nom est donné depuis la console de configuration Sage X3.

	Request config	Saisissez la chaîne de caractères de configuration si vous souhaitez récupérer les informations de débogage ou de traces. Par exemple, la chaîne de caractères peut être :
		
		`RequestConfigDebug="adxwss.trace.on=on";`
		
		Si vous devez utiliser plusieurs chaînes de caractères, séparez-les avec un &, par exemple,
		
		`RequestConfigDebug="adxwss .trace.on=on&adxwss.trace.size=16384";`
		
		Un outil tiers est nécessaire pour récupérer ce type d'informations.

	Publication name	Saisissez dans ce champ le nom de l'objet, de la liste ou du sous-programme publié(e) auquel/à laquelle vous souhaitez que votre *Studio Talend* accède.

	Action	Sélectionnez dans la liste l'action à effectuer. • **getAllDetails** : retourne les détails de tous les enregistrements. • **getSummary** : retourne le résumé de chaque enregistrement. • **getDetail** : retourne les détails des enregistrements répondant aux conditions configurées dans la table **Query condition**.
	Mapping	Remplissez ce tableau pour mapper les éléments variables de l'objet, le sous-programme ou la liste défini(e)(s) dans le serveur Web Sage X3 donné. Les colonnes à compléter incluent : **Column** : les colonnes définies dans l'éditeur de schéma de ce composant. **Group ID** : l'identifiant de chaque groupe d'éléments variables. Par exemple, un groupe d'éléments variables peut représenter l'un des attributs d'un objet. **Field name** : le nom de chaque élément variable.
	Conditions	Renseignez cette table afin de configurer la (les) condition(s) de requête. Les colonnes à renseigner comprennent : **Key** : les noms des éléments de variables utilisés comme clé pour l'extraction de données. **Value** : la valeur du champ de clé donné utilisé pour extraire les données correspondantes. Disponible lorsque l'option **getDetail** est sélectionnée dans la liste **Action**.
	Query condition	Cochez cette case pour paramétrer la (les) condition(s) de la requête. Les colonnes à compléter incluent : **Key** : le nom des éléments variables utilisés comme clé de l'extraction de données. **Value** : la valeur du champ clé donné utilisé pour extraire les données correspondantes. Indisponible lorsque l'option **getDetail** est sélectionnée dans la liste **Action**.
	Limit	Saisissez un chiffre ou un nombre indiquant le maximum de lignes de données à extraire. Indisponible lorsque l'option **getDetail** est sélectionnée dans la liste **Action**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Généralement utilisé comme composant de début. Un composant de sortie est nécessaire.	

Scénario : Utiliser les clés des requêtes pour extraire des données d'un système Sage X3 donné

Ce scénario décrit un Job à deux composants utilisé pour extraire une ligne ou des données d'un système Sage X3 donné. La méthode des objets est appelée, les éléments variables de cet objet sont des attributs. Les données utilisées dans ce scénario proviennent de l'exemple fourni par Sage X3.

Configurer le Job

1. Déposez un composant **tSageX3Input** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Reliez le composant **tSageX3Input** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer le schéma du tSageX3Input

1. Double-cliquez sur le **tSageX3Input** afin de configurer ses **Basic settings** dans la vue **Component**.

2. Dans la liste **Schema**, sélectionnez **Built-In** puis cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

3. Dans cet éditeur, cliquez douze fois sur le bouton **[+]** pour ajouter 12 lignes à la table.
4. Saisissez les noms que vous souhaitez utiliser pour chaque ligne. Dans cet exemple, ces lignes sont nommées d'après les noms de publication des attributs des objets définis dans le serveur Web Sage X3. Ces colonnes sont utilisées pour mapper les champs des attributs correspondants dans le système Sage X3.
5. Dans la colonne **Type**, cliquez sur la ligne *IMG* pour afficher une liste déroulante.
6. Dans la liste déroulante, sélectionnez **List**, puisque l'attribut apparaît plusieurs fois dans Sage. Répétez l'opération pour changer le type des lignes *TIT2NBLIG*, *ITMLNK* et *ZITMLNK* en **List** pour la même raison.
7. Cliquez sur **OK** pour valider les modifications et acceptez la propagation proposée par la boîte de dialogue qui s'ouvre.

Configurer la connexion au serveur Web Sage X3

1. Dans le champ **Endpoint address**, saisissez l'URL du service Web fournie par le serveur Web Sage X3. Dans cet exemple, l'URL est : *http://10.42.20.168:28880/adxwsvc/services/CAdxWebServiceXmlCC*.
2. Dans le champ **User**, saisissez le nom de l'utilisateur de Sage X3. Dans cet exemple, l'utilisateur est *ERP*.
3. Dans le champ **Language**, saisissez le nom du code langue X3 utilisé pour le démarrage d'un groupe de connexions, *FRA* dans cet exemple.
4. Dans le champ **Pool alias**, saisissez le nom du pool de connexion à utiliser. Dans cet exemple, le pool de connexion est appelé *TALEND*.
5. Dans le champ **Publication name**, saisissez le nom de publication de l'objet à appeler. Dans ce scénario, le nom de publication est *ITMDET*.

Paramétrer le mapping et configurer la condition de la requête

1. Dans les colonnes **Group ID** et **Field name** de la table **Mapping**, saisissez les valeurs correspondant à l'ID du groupe d'attributs et au nom de publication définis dans le serveur Web Sage X3. Dans cet exemple, les valeurs sont présentées comme suit :

Column	Group ID	Field Name
ITMREF	"TIT0_1"	"ITMREF"
ZITMREF	"TIT0_1"	"ZITMREF"
INTDES1	"TIT0_1"	"INTDES1"
ENAFLG	"TIT0_1"	"ENAFLG"
VIG	"TIT3_1"	"VIG"
IMG	"TIT3_1"	"IMG"
TIT2NBLIG	"TIT2_1"	"TIT2~NBLIG"
ITMLNK	"TIT2_1"	"ITMLNK"
ZITMLNK	"TIT2_1"	"ZITMLNK"
TEXTE	"TIT1_2"	"TEXTE"
WW_MODSTAMP	"ADXTEC"	"WW_MODSTA...
WW_MODUSER	"ADXTEC"	"WW_MODUSER"

Dans la table **Mapping**, la colonne **Column** a été automatiquement renseignée avec les colonnes que vous avez créées dans l'éditeur du schéma.

2. Cochez la case **Query condition** pour activer le tableau **Conditions**.
3. Sous le tableau **Conditions**, cliquez sur le bouton **[+]** pour ajouter une ligne.
4. Dans la colonne **Key**, saisissez le nom de publication associé à l'attribut de l'objet dont vous souhaitez extraire les données.
5. Dans la colonne **Value**, saisissez la valeur de l'attribut que vous avez sélectionné comme clé de l'extraction. Dans ce scénario, sa valeur est *CONTS00059*, une des références produit.

Exécuter le Job

1. Sélectionnez **Built-In** dans la liste **Schema** puis cliquez sur **[...]** à côté du champ **Edit schema** afin d'ouvrir l'éditeur du schéma.

2. Appuyez sur **F6** pour exécuter le Job.

   Les résultats sont affichés dans la vue **Run** :

```
.----------+----------------------+----------------------+------+------------------+--------------------+---------+-------------+
|                                                                                     tLogRow_1
|=---------+----------------------+----------------------+------+------------------+--------------------+---------+-------------+
|ITMREF    |ZITMREF               |INTDES1               |ENAFLG|VIG               |IMG                 |TIT2NBLIG|ITMLNK       |
|=---------+----------------------+----------------------+------+------------------+--------------------+---------+-------------+
|CONTS00060|Ecran 24" premium 16/10|Ecran 24" standard 28/10|2     |vig_CONTS00060.jpg|[img1_CONTS00060.jpg]|[1]      |[PFSER00097]|
'----------+----------------------+----------------------+------+------------------+--------------------+---------+-------------+
```

tSageX3Output

Propriétés du tSageX3Output

Famille de composant	Business/Sage X3	
Fonction	Le composant **tSageX3Output** se connecte au service Web fourni par un serveur Web Sage X3 donné et à partir de là, insère, met à jour ou supprime des données dans le système Sage X3 (serveur X3).	
Objectif	Ce composant écrit des données dans un système Sage X3 donné.	
Basic settings	*Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Endpoint address	Saisissez l'adresse du Service Web fourni par le serveur Web Sage X3 donné.
	Username et *Password*	Saisissez les informations d'authentification pour le Service Web, définies lors de la configuration du Serveur Web Sage X3. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Language	Saisissez le nom du code langue X3 utilisé pour le démarrage du groupe de connexions.
	Pool alias	Saisissez le nom du pool de connexion distribuant les requêtes reçues aux connexions disponibles. Ce nom est donné depuis la console de configuration Sage X3.

	Request config	Saisissez la chaîne de caractères de configuration si vous souhaitez récupérer les informations de débogage ou de traces. Par exemple, la chaîne de caractères peut être :
		
		`RequestConfigDebug="adxwss.trace.on=on";`
		
		Si vous devez utiliser plusieurs chaînes de caractères, séparez-les avec un &, par exemple,
		
		`RequestConfigDebug="adxwss .trace.on=on&adxwss.trace.size=16384";`

		Un outil tiers est nécessaire pour récupérer ce type d'informations.
	Publication name	Saisissez dans ce champ le nom de l'objet, de la liste ou du sous-programme publié(e) auquel/à laquelle vous souhaitez que votre *Studio Talend* accède.
	Action	Vous pouvez effectuer l'une des opérations suivantes sur les données d'un système Sage X3: **Insert** : insérer des données, **Update** : mettre à jour des données, **Delete** : supprimer des données.
	Mapping	Remplissez ce tableau pour mapper les éléments variables de l'objet, le sous-programme ou la liste défini(e)(s) dans le serveur Web Sage X3 donné au(x)quel(s)/à laquelle votre *Studio Talend* accède. Seuls les éléments sur lesquels vous souhaitez effectuer l'action sont sélectionnés et saisis dans le but de réaliser le mapping. Les colonnes à compléter incluent : **Column** : les colonnes définies dans l'éditeur du schéma de ce composant. **Key** : l'élément variable utilisé comme clé de l'insertion, de la mise à jour ou de la suppression de données. Cochez la case correspondante si la clé est un élément variable. **Group ID** : l'identifiant de chaque groupe d'éléments variables. Par exemple, un groupe d'éléments variables peut représenter l'un des attributs d'un objet. **Field name** : le nom du champ de chaque élément variable sélectionné.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de sortie et requiert un composant d'entrée.	

Scénario : Utiliser un service Web Sage X3 pour insérer des données dans un système Sage X3 donné

Ce scénario décrit un Job à deux composants, utilisé pour générer une ligne de données et insérer ces données dans un système Sage X3 donné. Vous pouvez trouver les données utilisées dans ce scénario dans l'exemple fourni par Sage X3. Le service Web Sage X3 est utilisé pour accéder à un objet.

Configurer le Job

1. Déposez un composant **tFixedFlowInput** et un **tSageX3Output** de la **Palette** dans l'espace de modélisation graphique.
2. Reliez le composant **tFixedFlowInput** au composant **tSageX3Output** à l'aide d'un lien **Row** > **Main**.

Configurer le schéma des données d'entrée

1. Double-cliquez sur le **tFixedFlowInput** afin d'afficher sa vue **Component** et définir ses propriétés.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir l'éditeur du schéma.

3. Cliquez quatre fois sur le bouton **[+]** afin d'ajouter quatre lignes au schéma.
4. Cliquez sur **OK** pour valider les modifications, puis acceptez la propagation proposée par la boîte de dialogue qui s'ouvre ensuite. Les quatre lignes apparaissent automatiquement dans la table **Values** de la vue **Component**.

5. Dans le tableau **Values** de la zone **Mode**, saisissez les valeurs de chacune des quatre lignes de la colonne **Value**. Dans ce scénario, les valeurs (de haut en bas) sont : *CONTS00059*, *Screen 24\" standard 16/10*, *Screen 24\" standard 28/10* et *2*.

*Les valeurs dans la colonne **Value** doivent être mises entre guillemets.*

Configurer la connexion au serveur Web Sage X3

1. Double-cliquez sur le composant **tSageX3Output** afin d'afficher sa vue **Component** et définir ses propriétés.

2. Dans le champ **Endpoint address**, saisissez l'URL du service Web fourni par le serveur Web Sage X3. Dans cet exemple, l'URL est : *http://10.42.20.168:28880/adxwsvc/services/CAdxWebServiceXmlCC*.

3. Dans le champ **User**, saisissez le nom de l'utilisateur de Sage X3. Dans cet exemple, l'utilisateur est *ERP*.

4. Dans le champ **Language**, saisissez le nom du code langue X3 utilisé pour démarrer un groupe de connexions. Dans cet exemple, le code est *FRA*.

5. Dans le champ **Pool alias**, saisissez le nom du pool de connexion à utiliser. Dans cet exemple, le pool de connexion est appelé *TALEND*.

6. Dans le champ **Publication name**, saisissez le nom de publication de l'objet à appeler. Dans ce scénario, le nom de publication est *ITMDET*.

7. Dans le champ **Action**, sélectionnez **insert** dans la liste déroulante.

Paramétrer le mapping

1. Dans la colonne **Field name** de la table **Mapping**, saisissez le nom des champs des attributs sur lesquels effectuer l'action sélectionnée.

2. Dans la colonne **Group ID** de la table **Mapping**, saisissez les valeurs correspondantes afin de regrouper les IDs des attributs sélectionnés. Ces IDs sont définis dans le serveur Web Sage X3.

Column	Key	Group ID	Field Name
newColumn		"TIT0_1"	"ITMREF"
newColumn1		"TIT0_1"	"ZITMREF"
newColumn2		"TIT0_1"	"INTDES1"
newColumn3		"TIT0_1"	"ENAFLG"

Dans la table **Mapping**, la colonne **Column** a été automatiquement renseignée avec toutes les colonnes récupérées du schéma du composant précédent.

Exécuter le Job

Appuyez sur **Ctrl+Espace** afin de sauvegarder votre Job puis sur **F6** pour exécuter le Job.

Pour vérifier les données que vous avez insérées dans ce scénario, vous pouvez utiliser le composant **tSageX3Input** pour lire les données concernées dans le serveur Sage X3.

Pour plus d'informations concernant l'utilisation du composant **tSageX3Input** pour lire des données, consultez le composant **tSageX3Input**.

tSalesforceBulkExec

Les composants **tSalesforceOutputBulk** et **tSalesforceBulkExec** sont généralement utilisés ensemble pour générer en sortie le fichier défini et exécuter ensuite les actions souhaitées sur votre fichier, dans votre Salesforce.com. Cette exécution en deux étapes est unifiée dans le composant **tSalesforceOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Propriétés du tSalesforceBulkExec

Famille de composant	Business/Cloud	
Fonction	Le composant **tSalesforceBulkExec** exécute les actions définies sur les données de chargement de masse préparées.	
Objectif	En tant que composant dédié, le **tSalesforceBulkExec** vous permet d'optimiser les performances lors des exécutions d'opérations définies dans votre Salesforce.com.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Login Type	Deux options sont disponibles : **Basic** : sélectionnez cette option afin de vous connecter à Salesforce.com en saisissant vos identifiants (nom d'utilisateur et mot de passe) dans le **tSalesforceConnection**. **OAuth2** : sélectionnez cette option pour accéder à Salesforce.com en saisissant votre Consumer key/Consumer Secret dans le **tSalesforceConnection**. Ainsi, vos identifiants ne sont pas visibles dans le **tSalesforceConnection**, mais un effort supplémentaire est requis : • Définissez une application connectée (Connected App) sur Salesforce.com pour obtenir votre Consumer key/Consumer Secret. Pour plus d'informations concernant les applications connectées, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais). • Lors de l'exécution d'un Job, la console du Studio affiche l'URL à traiter via OAuth. Pour plus d'informations, consultez *Scénario : Utiliser OAuth2 pour vous connecter à Salesforce.com*.
	Salesforce Webservice URL	Saisissez l'URL du service Web permettant de se connecter à la base de données Salesforce.
	Salesforce Version	Saisissez la version de Salesforce que vous utilisez.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.

	Consumer Key et *Consumer Secret*	Saisissez les détails de votre authentification OAuth. Ces informations sont disponibles dans la zone **OAuth Settings** de l'application connectée créée sur le site Salesforce.com. Pour saisir le Secret du consumer, cliquez sur le bouton **[...]** à côté du champ **Consumer Secret**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le Secret du consumer entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour comprendre ce qu'est une application connectée, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais).
	Callback Host et *Callback Port*	Saisissez l'URL de votre fonction de rappel pour l'authentification OAuth. Cette URL (contenant l'hôte et le port) est définie durant la création d'une application connectée et apparaît dans la zone **OAuth Settings** de l'application connectée.
	Token File	Saisissez le nom du jeton de fichier. Il stocke le jeton de rafraîchissement utilisé pour récupérer le jeton d'accès sans autorisation.
	Salesforce Version	Saisissez la version de Salesforce que vous utilisez.
	Bulk file path	Répertoire où sont stockées les données de masse que vous devez traiter.
	Action	Vous pouvez effectuer les opérations suivantes sur les données de l'objet Salesforce : **Insert** : insérer des données. **Update** : mettre à jour des données. **Upsert** : mettre à jour et insérer des données.
	Upsert Key Column	Spécifiez la colonne de clé pour l'opération d'upsert. Disponible lorsque l'option **Upsert** et sélectionnée dans la liste **Action**.
	Module	Sélectionnez l'objet adéquat dans la liste. **Custom Object** : cette option fait apparaître le champ **Custom Module Name** dans lequel vous pouvez saisir l'objet auquel vous souhaitez vous connecter.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

Advanced settings	*Rows to commit*	Saisissez le nombre de lignes devant être traitées par lot de données.
	Bytes to commit	Saisissez le nombre d'octets par lot de données devant être traités.
	Concurrency mode	Mode de simultanéité pour le Job : **Parallel** : Traite les lots en mode parallèle. **Serial** : Traite les lots en mode séquentiel.
	Wait time for checking batch state(milliseconds)	Saisissez l'intervalle de temps entre les vérifications de traitement des lots de données d'un Job jusqu'à ce que tous les lots soient traités.
	Use Socks Proxy	Cochez cette case si vous souhaitez utiliser un proxy. Une fois la case cochée, vous devez saisir les paramètres de connexion dans les champs qui apparaissent, l'hôte, le port, le nom d'utilisateur et le mot de passe du proxy que vous souhaitez utiliser.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_SUCCESS** : nombre de lignes traitées avec succès. Cette variable est une variable **After** et retourne un nombre entier. **NB_REJECT** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Les données de masse à traiter doivent être au format *.csv*. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez le composant **tSalesforceOutputBulk**.

tSalesforceConnection

Propriétés du tSalesforceConnection

Famille de composant	Business/Cloud	
Fonction	Le composant **tSalesforceConnection** ouvre une connexion vers un système Salesforce afin d'effectuer une transaction.	
Objectif	Ce composant permet de vous connecter à Salesforce.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. **Built-in** : Propriétés utilisées ponctuellement. **Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
	For salesforce bulk component	Cochez cette case si vous utilisez des composants de traitement de données de masse de la famille Salesforce. Une fois cochée, le champ **Salesforce Version** apparaît. Saisissez la version de Salesforce que vous utilisez. Pour plus d'informations concernant ces composants de traitement de données de masse, consultez *tSalesforceOutputBulk*, *tSalesforceBulkExec* et *tSalesforceOutputBulkExec*.
	Login Type	Deux options sont disponibles : **Basic** : sélectionnez cette option afin de vous connecter à Salesforce.com en saisissant vos identifiants (nom d'utilisateur et mot de passe) dans le **tSalesforceConnection**. **OAuth2** : sélectionnez cette option pour accéder à Salesforce.com en saisissant votre Consumer key/Consumer Secret dans le **tSalesforceConnection**. Ainsi, vos identifiants ne sont pas visibles dans le **tSalesforceConnection**, mais un effort supplémentaire est requis : • Définissez une application connectée (Connected App) sur Salesforce.com pour obtenir votre Consumer key/Consumer Secret. Pour plus d'informations concernant les applications connectées, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais). • Lors de l'exécution d'un Job, la console du Studio affiche l'URL à traiter via OAuth. Pour plus d'informations, consultez *Scénario : Utiliser OAuth2 pour vous connecter à Salesforce.com*.
	Salesforce Webservice URL	Saisissez l'URL du service Web permettant de se connecter à la base de données Salesforce.
	Salesforce Version	Saisissez la version de Salesforce que vous utilisez.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web.

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Consumer Key et *Consumer Secret*	Saisissez les détails de votre authentification OAuth. Ces informations sont disponibles dans la zone **OAuth Settings** de l'application connectée créée sur le site Salesforce.com. Pour saisir le Secret du consumer, cliquez sur le bouton **[...]** à côté du champ **Consumer Secret**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le Secret du consumer entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour comprendre ce qu'est une application connectée, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais).
	Callback Host et *Callback Port*	Saisissez l'URL de votre fonction de rappel pour l'authentification OAuth. Cette URL (contenant l'hôte et le port) est définie durant la création d'une application connectée et apparaît dans la zone **OAuth Settings** de l'application connectée.
	Token File	Saisissez le nom du jeton de fichier. Il stocke le jeton de rafraîchissement utilisé pour récupérer le jeton d'accès sans autorisation.
	Use Soap Compression	Cochez cette case pour activer la compression SOAP. La compression des messages SOAP permet un gain de performance.
	Timeout (milliseconds)	Saisissez le délai avant suspension de la requête sur Salesforce.com.
	Output Http Trace Message	Sélectionnez cette option pour écrire en sortie les interactions HTTP dans la console du Studio. Disponible lorsque l'option **For salesforce bulk component** est sélectionnée.
	Use Socks Proxy	Cochez cette case si vous souhaitez utiliser un proxy. Une fois la case cochée, vous devez saisir les paramètres de connexion dans les champs qui apparaissent, l'hôte, le port, le nom d'utilisateur et le mot de passe du proxy que vous souhaitez utiliser.
Advanced settings	*Client ID*	Définissez l'ID de l'utilisateur réel afin de différencier ceux qui utilisent le même compte et le même mot de passe pour accéder au site Web Salesforce.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec les composants Salesforce.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Utiliser OAuth2 pour vous connecter à Salesforce.com

Dans ce scénario, la méthode **OAuth2** (Open Authorization 2.0) est adoptée pour autoriser **tSalesforceConnection** à accéder aux ressources de Salesforce.com. Ainsi, la sécurité est optimisée et les utilisateurs n'ont pas besoin de saisir leur identifiant et mot de passe Salesforce directement dans le **tSalesforceConnection**, nécessaire pour le type d'authentification **Basic**.

Relier les composants

1. Déposez un **tSalesforceConnection**, un **tFixedFlowInput** et un **tSalesforceOutput** de la **Palette** dans l'espace de modélisation graphique.

2. Reliez le **tSalesforceConnection** au **tFixedFlowInput** à l'aide d'un lien **OnSubjobOk**.

3. Reliez le **tFixedFlowInput** au **tSalesforceOutput** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tSalesforceConnection** pour ouvrir sa vue **Basic settings**.

2. Dans la liste **Login Type**, sélectionnez **OAuth2**.
3. Dans les champs **Consumer Key** et **Consumer Secret**, saisissez les informations requises.
4. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

Sélectionnez **Use Inline Content (delimited file)** dans la zone **Mode**.

5. Dans le champ **Content**, saisissez les données à écrire dans Saleforce.com, par exemple :

```
Talend
```

6. Double-cliquez sur le **tSalesforceOutput** pour ouvrir sa vue **Basic settings**.

7. Cochez la case **Use an existing connection**.

8. Dans la liste **Action**, sélectionnez **insert** pour insérer le nom du compte, *Talend*.
9. Dans la liste **Module**, sélectionnez **Account**.
10. Cliquez sur le bouton **Edit schema** pour ouvrir l'éditeur de schéma.

11. Dans le panneau de droite, supprimez toutes les colonnes, sauf *Name*.
12. Cliquez sur le bouton pour copier la colonne *Name* de droite à gauche.
13. Cliquez sur **OK** pour valider les paramètres et fermer l'éditeur.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job. La console du Studio affiche l'URL (surlignée en jaune) de l'autorisation OAuth.

3. Copiez l'URL dans la barre d'adresse de votre navigateur. La page de connexion à Salesforce.com s'ouvre.

4. Saisissez votre identifiant et votre mot de passe puis cliquez sur le bouton **Log in to Salesforce**.

   La réponse à l'authentification s'affiche dans le navigateur.

5. Le sous-job s'exécute.

```
2013-09-23 10:33:53.265:INFO:oejs.Server:jetty-8.1.12.v20130726
[statistics] connecting to socket on port 3536
[statistics] connected
Paste this URL into a web browser to authorize BigQuery Access:
https://login.salesforce.com/services/oauth2/authorize?response
type=code&client_id=3MVG9A2kN3Bn17hspsdhvZoe_y2kf38K58tRWqdZT22
5PjLAIbBQEJJwjOs2Yuc_HpWccLFbtkzxv8kZU9kfQ&redirect_uri=https%3
%2F%2Flocalhost%3A8446
2013-09-23 10:33:53.475:INFO:oejus.SslContextFactory:Enabled
Protocols [SSLv2Hello, SSLv3, TLSv1] of [SSLv2Hello, SSLv3,
TLSv1]
2013-09-23 10:33:53.491:INFO:oejs.AbstractConnector:Started
SslSelectChannelConnector@0.0.0.0:8446
[statistics] disconnected
```

   Comme vous pouvez le constater dans la console, le Job est correctement exécuté.

6. Allez sur Salesforce.com et consultez la page Account module. Le nom de compte *Talend* est ajouté.

Account Detail Edit

Account Owner	andy zhang [Change]
Account Name	Talend [View Hierarchy]
Parent Account	
Account Number	
Account Site	
Type	
Industry	
Annual Revenue	
Billing Address	
Customer Priority	
SLA Expiration Date	
Number of Locations	
Active	
Created By	andy zhang, 9/22/2013 12:39 AM
Description	
Custom Links	Billing

tSalesforceGetDeleted

Propriétés du tSalesforceGetDeleted

Famille de composant	Business/Cloud	
Fonction	Le composant **tSalesforceGetDeleted** récupère les données supprimées d'un objet Salesforce sur une période donnée.	
Objectif	Ce composant récupère les données supprimées de l'objet Salesforce spécifié sur une période spécifiée.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Login Type	Deux options sont disponibles : **Basic** : sélectionnez cette option afin de vous connecter à Salesforce.com en saisissant vos identifiants (nom d'utilisateur et mot de passe) dans le **tSalesforceConnection**. **OAuth2** : sélectionnez cette option pour accéder à Salesforce.com en saisissant votre Consumer key/Consumer Secret dans le **tSalesforceConnection**. Ainsi, vos identifiants ne sont pas visibles dans le **tSalesforceConnection**, mais un effort supplémentaire est requis : • Définissez une application connectée (Connected App) sur Salesforce.com pour obtenir votre Consumer key/Consumer Secret. Pour plus d'informations concernant les applications connectées, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais). • Lors de l'exécution d'un Job, la console du Studio affiche l'URL à traiter via OAuth. Pour plus d'informations, consultez *Scénario : Utiliser OAuth2 pour vous connecter à Salesforce.com*.
	Salesforce Webservice URL	Saisissez l'URL du service Web permettant de se connecter à la base de données Salesforce.
	Salesforce Version	Saisissez la version de Salesforce que vous utilisez.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Consumer Key et *Consumer Secret*	Saisissez les détails de votre authentification OAuth. Ces informations sont disponibles dans la zone **OAuth Settings** de l'application connectée créée sur le site Salesforce.com. Pour saisir le Secret du consumer, cliquez sur le bouton **[...]** à côté du champ **Consumer Secret**, puis, dans la boîte de dialogue qui

		s'ouvre, saisissez le Secret du consumer entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour comprendre ce qu'est une application connectée, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais).
	Callback Host et *Callback Port*	Saisissez l'URL de votre fonction de rappel pour l'authentification OAuth. Cette URL (contenant l'hôte et le port) est définie durant la création d'une application connectée et apparaît dans la zone **OAuth Settings** de l'application connectée.
	Token File	Saisissez le nom du jeton de fichier. Il stocke le jeton de rafraîchissement utilisé pour récupérer le jeton d'accès sans autorisation.
	Timeout (milliseconds)	Saisissez le délai avant suspension de la requête sur Salesforce.com.
	Module	Sélectionnez l'objet adéquat dans la liste. **Custom Object** : cette option fait apparaître le champ **Costum Module Name** dans lequel vous pouvez saisir l'objet auquel vous souhaitez vous connecter.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
	Start Date	Saisissez entre guillemets la date à laquelle vous voulez commencer la recherche. Pour cela, utilisez le format de date suivant : "yyyy-MM-dd HH:mm:ss". Vous ne pouvez effectuer la recherche que sur les 30 derniers jours.
	End Date	Saisissez entre guillemet la date à laquelle vous voulez terminer la recherche. Pour cela, utilisez le format de date suivant : "yyyy-MM-dd HH:mm:ss".
Advanced settings	*Use Soap Compression*	Cochez cette case pour activer la compression SOAP. La compression des messages SOAP permet un gain de performance.
	Client ID	Définissez l'ID de l'utilisateur réel afin de différencier ceux qui utilisent le même compte et le même mot de passe pour accéder au site Web Salesforce.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Récupérer les données supprimées du serveur Salesforce

Ce scénario décrit un Job à deux composants permettant de récupérer les données supprimées du serveur Salesforce, dans les 5 derniers jours.

Configurer le Job

1. Cliquez et déposez les composants **tSalesforceGetDeleted** et **tLogRow** de la **Palette** dans le Job Designer.
2. Reliez-les via un lien de type **Row** > **Main**.

Configurer la connexion au serveur Salesforce

1. Double-cliquez sur le composant **tSalesforceGetDeleted** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés :

2. Dans le champ **Salesforce WebService URL**, laissez l'URL du service Web Salesforce par défaut ou saisissez l'URL à laquelle vous souhaitez accéder.

3. Dans les champs **Username** et **Password**, saisissez votre identifiant et votre mot de passe de connexion au service Web.

4. Dans la liste **Module**, sélectionnez l'objet auquel vous souhaitez accéder. Dans ce scénario, l'objectif est d'accéder à l'objet **Account**.

Paramétrer les conditions de la recherche

1. Dans la liste **Schema**, sélectionnez **Repository** puis cliquez sur le bouton **[...]** pour sélectionner le schéma du Repository que vous souhaitez utiliser pour le composant. Si vous n'avez pas enregistré le schéma dans les métadonnées du Repository, sélectionnez **Built-in** dans la liste **Schema** et cliquez sur le bouton **[...]** à côté du champ **Edit schema** et paramétrez le schéma manuellement.

2. Dans les champs **Start Date** et **End Date**, saisissez la date de début et de fin de recherche des données supprimées en utilisant le format de date suivant : "*yyyy-MM-dd HH:mm:ss*". Vous ne pouvez effectuer la recherche que sur les 30 derniers jours. Dans ce scénario, l'objectif est de chercher les données supprimées dans les cinq derniers jours.

Exécuter le Job

1. Double-cliquez sur le composant **tLogRow** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés :

2. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

3. Dans la zone **Mode**, sélectionnez l'option **Vertical** pour faire apparaître le résultat dans sous forme de tableau dans la console.

4. Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job tSalesforceGetDeleted_scenario at 17:17 25/06/2009.

.-----------------------------------------------------.
|                    #1. tLogRow_1                    |
+----------------------+-----------------------------+
| key                  | value                       |
+----------------------+-----------------------------+
| Id                   | 001700000003smNAAR          |
| IsDeleted            | true                        |
| MasterRecordId       | null                        |
| Name                 | sForce                      |
| Type                 | null                        |
| ParentId             | null                        |
| BillingStreet        | The Landmark at One Market  |
| BillingCity          | San Francisco               |
| BillingState         | CA                          |
| BillingPostalCode    | 94087                       |
| BillingCountry       | US                          |
| ShippingStreet       | null                        |
| ShippingCity         | null                        |
| ShippingState        | null                        |
| ShippingPostalCode   | null                        |
| ShippingCountry      | null                        |
| Phone                | (415) 901-7000              |
| Fax                  | (415) 901-7002              |
| AccountNumber        | CX355105                    |
| Website              | www.sforce.com              |
| Sic                  | null                        |
| Industry             | null                        |
| AnnualRevenue        | 0.0                         |
| NumberOfEmployees    | null                        |
| Ownership            | null                        |
| TickerSymbol         | null                        |
| Description          | null                        |
| Rating               | null                        |
```

La console affiche les données supprimées récupérées par le composant **tSalesforceGetDeleted** sous forme de tableau.

tSalesforceGetServerTimestamp

Propriétés du tSalesforceGetServerTimestamp

Famille de composant	Business/Cloud	
Fonction	Le composant **tSalesforceGetServerTimestamp** récupère la date actuelle du serveur Salesforce.	
Objectif	Ce composant récupère la date actuelle sous forme de timestamp du serveur Salesforce.	
Basic settings	*Use an existing connection*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
	Login Type	Deux options sont disponibles : **Basic** : sélectionnez cette option afin de vous connecter à Salesforce.com en saisissant vos identifiants (nom d'utilisateur et mot de passe) dans le **tSalesforceConnection**. **OAuth2** : sélectionnez cette option pour accéder à Salesforce.com en saisissant votre Consumer key/Consumer Secret dans le **tSalesforceConnection**. Ainsi, vos identifiants ne sont pas visibles dans le **tSalesforceConnection**, mais un effort supplémentaire est requis : • Définissez une application connectée (Connected App) sur Salesforce.com pour obtenir votre Consumer key/Consumer Secret. Pour plus d'informations concernant les applications connectées, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais). • Lors de l'exécution d'un Job, la console du Studio affiche l'URL à traiter via OAuth. Pour plus d'informations, consultez *Scénario : Utiliser OAuth2 pour vous connecter à Salesforce.com*.
	Salesforce Webservice URL	Saisissez l'URL du service Web permettant de se connecter à la base de données Salesforce.

	Salesforce Version	Saisissez la version de Salesforce que vous utilisez.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Consumer Key et *Consumer Secret*	Saisissez les détails de votre authentification OAuth. Ces informations sont disponibles dans la zone **OAuth Settings** de l'application connectée créée sur le site Salesforce.com. Pour saisir le Secret du consumer, cliquez sur le bouton **[...]** à côté du champ **Consumer Secret**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le Secret du consumer entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour comprendre ce qu'est une application connectée, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais).
	Callback Host et *Callback Port*	Saisissez l'URL de votre fonction de rappel pour l'authentification OAuth. Cette URL (contenant l'hôte et le port) est définie durant la création d'une application connectée et apparaît dans la zone **OAuth Settings** de l'application connectée.
	Token File	Saisissez le nom du jeton de fichier. Il stocke le jeton de rafraîchissement utilisé pour récupérer le jeton d'accès sans autorisation.
	Timeout (milliseconds)	Saisissez le délai avant suspension de la requête sur Salesforce.com.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
Advanced settings	*Use Socks Proxy*	Cochez cette case si vous souhaitez utiliser un proxy. Une fois la case cochée, vous devez saisir les paramètres de connexion dans les champs qui apparaissent, l'hôte, le port, le nom d'utilisateur et le mot de passe du proxy que vous souhaitez utiliser.
	Use Soap Compression	Cochez cette case pour activer la compression SOAP. La compression des messages SOAP permet un gain de performance.

	Client ID	Définissez l'ID de l'utilisateur réel afin de différencier ceux qui utilisent le même compte et le même mot de passe pour accéder au site Web Salesforce.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSalesforceGetUpdated

Propriétés du tSalesforceGetUpdated

Famille de composant	Business/Cloud	
Fonction	Le composant **tSalesforceGetUpdated** récupère les données mises à jour d'un objet Salesforce sur une période donnée.	
Objectif	Ce composant récupère les données mises à jour de l'objet Salesforce spécifié sur une période spécifiée.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Login Type	Deux options sont disponibles : **Basic** : sélectionnez cette option afin de vous connecter à Salesforce.com en saisissant vos identifiants (nom d'utilisateur et mot de passe) dans le **tSalesforceConnection**. **OAuth2** : sélectionnez cette option pour accéder à Salesforce.com en saisissant votre Consumer key/Consumer Secret dans le **tSalesforceConnection**. Ainsi, vos identifiants ne sont pas visibles dans le **tSalesforceConnection**, mais un effort supplémentaire est requis : • Définissez une application connectée (Connected App) sur Salesforce.com pour obtenir votre Consumer key/Consumer Secret. Pour plus d'informations concernant les applications connectées, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais). • Lors de l'exécution d'un Job, la console du Studio affiche l'URL à traiter via OAuth. Pour plus d'informations, consultez *Scénario : Utiliser OAuth2 pour vous connecter à Salesforce.com*.
	Salesforce Webservice URL	Saisissez l'URL du service Web permettant de se connecter à la base de données Salesforce.
	Salesforce Version	Saisissez la version de Salesforce que vous utilisez.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Consumer Key et *Consumer Secret*	Saisissez les détails de votre authentification OAuth. Ces informations sont disponibles dans la zone **OAuth Settings** de l'application connectée créée sur le site Salesforce.com. Pour saisir le Secret du consumer, cliquez sur le bouton **[...]** à côté du champ **Consumer Secret**, puis, dans la boîte de dialogue qui

		s'ouvre, saisissez le Secret du consumer entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour comprendre ce qu'est une application connectée, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais).
	Callback Host et *Callback Port*	Saisissez l'URL de votre fonction de rappel pour l'authentification OAuth. Cette URL (contenant l'hôte et le port) est définie durant la création d'une application connectée et apparaît dans la zone **OAuth Settings** de l'application connectée.
	Token File	Saisissez le nom du jeton de fichier. Il stocke le jeton de rafraîchissement utilisé pour récupérer le jeton d'accès sans autorisation.
	Timeout (milliseconds)	Saisissez le délai avant suspension de la requête sur Salesforce.com.
	Module	Sélectionnez l'objet adéquat dans la liste. **Custom Object** : cette option fait apparaître le champ **Custom Module Name** dans lequel vous pouvez saisir l'objet auquel vous souhaitez vous connecter.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
	Start Date	Saisissez entre guillemet la date à laquelle vous voulez commencer la recherche. Pour cela, utilisez le modèle de date suivant : "yyyy-MM-dd HH:mm:ss". Vous ne pouvez effectuer la recherche que sur les 30 derniers jours.
	End Date	Saisissez entre guillemets la date à laquelle vous voulez terminer la recherche. Pour cela, utilisez le modèle de date suivant : "yyyy-MM-dd HH:mm:ss".
Advanced settings	*Use Soap Compression*	Cochez cette case pour activer la compression SOAP. La compression des messages SOAP permet un gain de performance.
	Client ID	Définissez l'ID de l'utilisateur réel afin de différencier ceux qui utilisent le même compte et le même mot de passe pour accéder au site Web Salesforce.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSalesforceInput

Propriétés du tSalesforceInput

Famille de composant	Business/Cloud	
Fonction	Le composant **tSalesforceInput** se connecte à un objet de la base de données Salesforce via le service Web adéquat.	
Objectif	Ce composant permet d'extraire les données d'une base de données Salesforce à l'aide d'une requête.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant connexion de base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Query mode	Deux options sont disponibles : **Query** et **Bulk Query**.
	Login Type	Deux options sont disponibles : **Basic** : sélectionnez cette option afin de vous connecter à Salesforce.com en saisissant vos identifiants (nom d'utilisateur et mot de passe) dans le **tSalesforceConnection**. **OAuth2** : sélectionnez cette option pour accéder à Salesforce.com en saisissant votre Consumer key/Consumer Secret dans le **tSalesforceConnection**. Ainsi, vos identifiants ne sont pas visibles dans le **tSalesforceConnection**, mais un effort supplémentaire est requis : • Définissez une application connectée (Connected App) sur Salesforce.com pour obtenir votre Consumer key/Consumer Secret. Pour plus d'informations concernant les applications connectées, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais). • Lors de l'exécution d'un Job, la console du Studio affiche l'URL à traiter via OAuth. Pour plus d'informations, consultez *Scénario : Utiliser OAuth2 pour vous connecter à Salesforce.com*.
	Salesforce Webservice URL	Saisissez l'URL du service Web permettant de se connecter à la base de données Salesforce.
	Salesforce Version	Saisissez la version de Salesforce que vous utilisez.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web.

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Consumer Key et *Consumer Secret*	Saisissez les détails de votre authentification OAuth. Ces informations sont disponibles dans la zone **OAuth Settings** de l'application connectée créée sur le site Salesforce.com. Pour saisir le Secret du consumer, cliquez sur le bouton **[...]** à côté du champ **Consumer Secret**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le Secret du consumer entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour comprendre ce qu'est une application connectée, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais).
	Callback Host et *Callback Port*	Saisissez l'URL de votre fonction de rappel pour l'authentification OAuth. Cette URL (contenant l'hôte et le port) est définie durant la création d'une application connectée et apparaît dans la zone **OAuth Settings** de l'application connectée.
	Token File	Saisissez le nom du jeton de fichier. Il stocke le jeton de rafraîchissement utiliser pour récupérer le jeton d'accès sans autorisation.
	Timeout (milliseconds)	Saisissez le délai avant suspension de la requête sur Salesforce.com. Disponible lorsque l'option **Query** est sélectionnée dans la liste **Query mode**.
	Module	Sélectionnez l'objet adéquat dans la liste. **Custom Object** : cette option fait apparaître le champ **Costum Module Name** dans lequel vous pouvez saisir l'objet auquel vous souhaitez vous connecter.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Dans ce composant, le schéma est relatif au Module sélectionné. *Pour retrouver une colonne d'un module lié, il est nécessaire de définir la colonne, d'une manière précise, dans la vue Edit Schema, sinon la requête relationnelle ne fonctionnera pas. La syntaxe correcte est : `NameofCurrentModule_NameofLinkedModule_NameofColumnofInterest`.*
	Query condition	Saisissez la requête permettant de sélectionner les données à extraire. Exemple : `account_name= 'Talend'`
	Manual input of SOQL query	Cochez cette case pour saisir manuellement la requête Salesforce que vous souhaitez effectuer dans le champ **Query**.

	Query all records (include deleted records)	Cochez cette case pour interroger tous les enregistrements, y compris les enregistrements supprimés. Disponible lorsque l'option **Query** est sélectionnée dans la liste **Query mode**.
Advanced settings	*Batch Size*	Nombre d'enregistrements dans chaque lot récupéré. Disponible lorsque l'option **Query** est sélectionnée dans la liste **Query mode**.
	Use Socks Proxy	Cochez cette case si vous souhaitez utiliser un proxy. Une fois la case cochée, vous devez saisir les paramètres de connexion dans les champs qui apparaissent, l'hôte, le port, le nom d'utilisateur et le mot de passe du proxy que vous souhaitez utiliser.
	Normalize delimited (for child relationship)	Caractère, chaîne ou expression régulière utilisé pour normaliser les données collectées à l'aide de requêtes effectuées sur les relations hiérarchiques entre les différents objets Salesforce. Disponible lorsque l'option **Query** est sélectionnée dans la liste **Query mode**.
	Column name delimiter (for child relationship)	Caractère, chaîne ou expression régulière utilisé pour séparer le nom de l'objet parent du nom de l'objet fils lorsque vous effectuez une requête sur les relations hiérarchiques entre les différents objets Salesforce. Disponible lorsque l'option **Query** est sélectionnée dans la liste **Query mode**.
	Use Soap Compression	Cochez cette case pour activer la compression SOAP. La compression des messages SOAP permet un gain de performance, notamment lors des opérations de batch.
	Output Http Trace Message	Cochez cette case pour écrire en sortie le message HTTP Trace. Disponible lorsque l'option **Bulk Query** est sélectionnée dans la liste **Query mode**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Client ID	Définissez l'ID de l'utilisateur réel afin de différencier ceux qui utilisent le même compte et le même mot de passe pour accéder au site Web Salesforce. Disponible lorsque l'option **Query** est sélectionnée dans la liste **Query mode**.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Utiliser des requêtes pour extraire des données d'une base de données Salesforce

Ce scénario décrit un Job comprenant quatre composants, utilisés pour extraire des ensembles spécifiques de données d'objets parents et enfants dans une base de données Salesforce.

Configurer le Job

1. Déposez deux composants **tSalesforceInput** et deux **tLogRow** dans l'espace de modélisation graphique.
2. Reliez chaque composant **tSalesforceInput** à un **tLogRow**, à l'aide de liens **Row** > **Main**.
3. Reliez le **tSalesforceInput_1** au **tSalesforceInput_2** en utilisant un lien **OnSubjobOk**.

Configurer la connexion au serveur Salesforce pour l'objet parent

1. Double-cliquez sur le **tSalesforceInput_1** afin de définir ses **Basic settings** dans la vue **Component**.

2. Dans le champ **Salesforce WebService URL**, saisissez l'URL de la base de données à laquelle vous souhaitez vous connecter.

3. Renseignez vos informations d'authentification dans les champs correspondants, **Username** pour l'identifiant et **Password** pour le mot de passe.

4. Saisissez la limite de temps que vous souhaitez appliquer à la requête, dans le champ **Timeout (milliseconds)**.

Configurer la requête et le schéma pour l'objet parent

1. Sélectionnez le **Module** (objet Salesforce) que vous souhaitez interroger.

2. Cochez la case **Manual input of SOQL Query** puis saisir votre requête ou votre requête relationnelle, en respectant la syntaxe SOQL requise, dans le champ **Query**.

3. Sélectionnez **Built-In** dans la liste **Schema** puis cliquez sur le bouton **[...]** à côté du champ **Edit schema**, afin d'ouvrir l'éditeur du schéma.

Dans cet exemple, les colonnes *IsWon* et *FiscalYear* dans la requête sont situées dans le module **Opportunity** spécifié. La colonne *Name* est dans un module lié, appelé *Account*. Pour retourner une colonne d'un module lié, la syntaxe correcte est de saisir le nom du module lié, suivi par le caractère point (.), puis par le nom de la colonne appropriée. Ici, la requête requise est donc :

```
"SELECT IsWon, FiscalYear, Account.Name FROM Opportunity".
```

4. Ajoutez une nouvelle colonne pour les champs extraits de la colonne *Name* du module *Account*.

5. Appelez cette colonne *Opportunity_Account_Name* puis cliquez sur **OK** afin de sauvegarder les modifications et fermer l'éditeur du schéma. .

*Pour retrouver une colonne d'un modèle lié, il est nécessaire de définir la colonne d'une manière précise dans la vue **Edit Schema**. La syntaxe correcte est : NameofCurrentModule_NameofLinkedModule_NameofColumnofInterest. Dans notre exemple, la colonne doit se nommer : Opportunity_Account_Name. Si cette syntaxe n'est pas respectée, les données de la table liée ne seront pas retournées.*

Configurer la connexion au serveur Salesforce pour l'objet fils

1. Double-cliquez sur le **tSalesforceInput_2** afin de définir ses **Basic settings** dans la vue **Component**.

2. Dans le champ **Salesforce WebService URL**, saisissez l'URL de la base de données à laquelle vous souhaitez vous connecter.

3. Saisissez vos informations d'authentification dans les champs **Username** pour l'identifiant et **Password** pour le mot de passe.

4. Saisissez la limite de temps que vous souhaitez appliquer sur la requête, dans le champ **Timeout (milliseconds)**.

Configurer la requête et le schéma pour l'objet fils

1. Sélectionnez le **Module** (objet Salesforce) que vous souhaitez interroger.

2. Cochez la case **Manual input of SOQL Query** puis saisir votre requête ou votre requête relationnelle, en respectant la syntaxe SOQL requise, dans le champ **Query**.

   Dans cet exemple, l'objectif est d'extraire les champs *Id* et *CaseNumber* du module *Case*, ainsi que les champs *Name* du module *Account*. La requête est donc : `"SELECT Id, CaseNumber, Account.Name FROM Case"`.

3. Sélectionnez **Built-In** dans la liste **Schema** et cliquez sur le bouton **[...]** à côté du champ **Edit schema**, afin d'ouvrir l'éditeur du schéma.

4. Ajoutez une nouvelle colonne pour les champs extraits de la colonne *Name* du module *Account*.

5. Nommez cette colonne *Case_Account_Name* puis cliquez sur **OK** pour sauvegarder les modifications et fermer l'éditeur du schéma.

Exécuter le Job

1. Cliquez sur chaque composant **tLogRow** afin de définir les **Basic settings** comme nécessaire.

   Dans cet exemple, vous n'avez pas besoin de modifier les paramètres des **tLogRow**.

2. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

   Les résultats sont affichés dans la console de la vue **Run** :

```
|true|2007|GenePoint
|true|2007|Burlington Textiles Corp of America
|true|2007|United Oil & Gas Corp.
|false|2007|United Oil & Gas Corp.
|true|2007|United Oil & Gas Corp.
|false|2007|United Oil & Gas Corp.
|true|2007|United Oil & Gas Corp.
|true|2007|United Oil & Gas Corp.
|false|2007|GenePoint
|true|2007|GenePoint
|true|2007|Edge Communications
|false|2007|Edge Communications
|true|2007|Edge Communications
|false|2007|Pyramid Construction Inc.
|false|2007|Dickenson plc
|true|2007|Grand Hotels & Resorts Ltd
|true|2007|Grand Hotels & Resorts Ltd
|false|2007|Grand Hotels & Resorts Ltd
50070000005DBELAA4|00001000|Edge Communications
50070000005DBEMAA4|00001001|United Oil & Gas Corp.
50070000005DBENAA4|00001002|United Oil & Gas Corp.
50070000005DBEOAA4|00001003|Express Logistics and Transport
50070000005DBEPAA4|00001004|Express Logistics and Transport
50070000005DBEQAA4|00001005|Express Logistics and Transport
50070000005DBERAA4|00001006|GenePoint
50070000005DBESAA4|00001007|Grand Hotels & Resorts Ltd
50070000005DBETAA4|00001008|Grand Hotels & Resorts Ltd
50070000005DBEUAA4|00001009|United Oil & Gas, UK
50070000005DBEVAA4|00001010|United Oil & Gas, Singapore
```

tSalesforceOutput

Propriétés du tSalesforceOutput

Famille de composant	Business/Cloud	
Fonction	Le composant **tSalesforceOutput** écrit dans un objet de la base de données Salesforce via le service Web adéquat.	
Objectif	Ce composant permet d'écrire des données dans une base de données Salesforce.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant connexion de base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Login Type	Deux options sont disponibles : **Basic** : sélectionnez cette option afin de vous connecter à Salesforce.com en saisissant vos identifiants (nom d'utilisateur et mot de passe) dans le **tSalesforceConnection**. **OAuth2** : sélectionnez cette option pour accéder à Salesforce.com en saisissant votre Consumer key/Consumer Secret dans le **tSalesforceConnection**. Ainsi, vos identifiants ne sont pas visibles dans le **tSalesforceConnection**, mais un effort supplémentaire est requis : • Définissez une application connectée (Connected App) sur Salesforce.com pour obtenir votre Consumer key/Consumer Secret. Pour plus d'informations concernant les applications connectées, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais). • Lors de l'exécution d'un Job, la console du Studio affiche l'URL à traiter via OAuth. Pour plus d'informations, consultez *Scénario : Utiliser OAuth2 pour vous connecter à Salesforce.com*.
	Salesforce Webservice URL	Saisissez l'URL du service Web permettant de se connecter à la base de données Salesforce.

	Salesforce Version	Saisissez la version de Salesforce que vous utilisez.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Consumer Key et *Consumer Secret*	Saisissez les détails de votre authentification OAuth. Ces informations sont disponibles dans la zone **OAuth Settings** de l'application connectée créée sur le site Salesforce.com. Pour saisir le Secret du consumer, cliquez sur le bouton **[...]** à côté du champ **Consumer Secret**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le Secret du consumer entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour comprendre ce qu'est une application connectée, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais).
	Callback Host et *Callback Port*	Saisissez l'URL de votre fonction de rappel pour l'authentification OAuth. Cette URL (contenant l'hôte et le port) est définie durant la création d'une application connectée et apparaît dans la zone **OAuth Settings** de l'application connectée.
	Token File	Saisissez le nom du jeton de fichier. Il stocke le jeton de rafraîchissement utilisé pour récupérer le jeton d'accès sans autorisation.
	Timeout (milliseconds)	Saisissez le délai avant suspension de la requête sur Salesforce.com.
	Action	Vous pouvez effectuer l'une des opérations suivantes sur les données de l'objet Salesforce : **Insert** : insérer les données. **Update** : mettre les données à jour. **Delete** : supprimer les données. **Upsert** : mettre à jour et insérer les données.
	Upsert Key Column	Spécifiez la colonne de clé pour l'opération d'upsert. Disponible lorsque l'option **Upsert** et sélectionnée dans la liste **Action**.
	Module	Sélectionnez l'objet adéquat dans la liste. **Custom Object** : cette option fait apparaître le champ **Costum Module Name** dans lequel vous pouvez saisir l'objet auquel vous souhaitez vous connecter.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Extended Output*	Cette case est cochée par défaut. Elle permet de transférer les données de sortie en lot. Dans le champ **Rows to commit**, spécifiez le nombre de lignes par lot.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**. Le lien **Reject** est disponible uniquement lorsque les cases **Extended Output** et **Die on error** sont décochées.
	Error logging file	Renseignez ce champ si vous souhaitez créer un fichier contenant les logs d'erreur. Pour cela, cliquez sur le bouton **[...]** à côté de ce champ pour définir le chemin d'accès et le nom du fichier
	Use Socks Proxy	Cochez cette case si vous souhaitez utiliser un proxy. Une fois la case cochée, vous devez saisir les paramètres de connexion dans les champs qui apparaissent, l'hôte, le port, le nom d'utilisateur et le mot de passe du proxy que vous souhaitez utiliser.
	Ignore NULL fields values	Cochez cette case pour ignorer les valeurs nulles en mode **Update** ou **Upsert**.
	Use Soap Compression	Cochez cette case pour activer la compression SOAP. La compression des messages SOAP permet un gain de performance.
	Retrieve inserted ID	Cochez cette case afin de permettre à Salesforce.com de retourner l'ID Salesforce produit pour une nouvelle ligne devant être insérée. La colonne ID est ajoutée au schéma des données traitées dans Salesforce.com. Cette option est disponible uniquement si vous avez sélectionné l'action **Insert**, mais non en mode batch, c'est-à-dire sans l'option **Extended Output**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Client ID	Définissez l'ID de l'utilisateur réel afin de différencier ceux qui utilisent le même compte et le même mot de passe pour accéder au site Web Salesforce.
	Relationship mapping for upsert (pour **upsert** uniquement)	Cliquez sur le bouton **[+]** pour ajouter des lignes et spécifiez les champs d'IDs externes du flux d'entrée, les champs de relations lookup dans le module d'upsert, le module de lookup, ainsi que les champs d'ID externes dans le module de lookup. **Column name of Talend schema** : champ d'ID externe dans le flux d'entrée. **Lookup field name** : champ de relations lookup dans le module d'upsert. **Module name** : nom du module de lookup. **External id name** : champ d'ID externe dans le module de lookup. • **Column name of Talend schema** fait référence aux champs du schéma du composant précédent le **tSalesforceOutput**. Ces colonnes sont mises en correspondance par rapport aux champs d'IDs externes spécifiés dans la colonne **External id**

		name. Ces champs sont ceux spécifiés dans la colonne **Module name** du module de lookup. • **Lookup field name** fait référence aux champs de relations lookup du module sélectionné dans la liste **Module** de la vue **Basic settings**. Ils servent à établir les relations avec le module de lookup spécifié dans le champ **Module name**. • Pour plus d'informations concernant la définition des champs de relation lookup et comment saisir le bon nom dans le champ **Lookup field name**, rendez-vous sur le site Web de Salesforce et lancez l'application Salesforce Data Loader.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_SUCCESS** : nombre de lignes traitées avec succès. Cette variable est une variable **After** et retourne un nombre entier. **NB_REJECT** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Supprimer des données de l'objet Account

Ce scénario décrit un Job à deux composants permettant de supprimer une entrée de l'objet Account.

Déposer et relier les composants

1. Cliquez et déposez les composants **tSalesforceInput** et **tSalesforceOutput** de la **Palette** dans le Job Designer.
2. Reliez-les via un lien de type **Row** > **Main**.

Effectuer une requête sur le contenu à effacer

1. Double-cliquez sur le composant **tSalesforceInput** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés :

2. Dans la liste **Property Type**, sélectionnez **Repository** si vous avez stocké la connexion au serveur Salesforce dans les métadonnées du **Repository**. Les autres champs seront renseignés automatiquement. Si vous n'avez pas stocké la connexion dans les métadonnées, sélectionnez **Built-in** dans la liste et renseignez les champs manuellement.

   Pour plus d'informations concernant la création de métadonnées Salesforce, consultez le *Guide utilisateur du Studio Talend*.

3. Dans le champ **Salesforce WebService URL**, laissez l'URL du service Web Salesforce par défaut ou saisissez l'URL à laquelle vous souhaitez accéder.

4. Dans les champs **Username** et **Password**, saisissez votre identifiant et votre mot de passe de connexion au service Web.

5. Saisissez, en millisecondes, le délai avant suspension dans le champ **Timeout (milliseconds)**. Dans ce scénario, utilisez la valeur par défaut.

6. Dans la liste **Module**, sélectionnez l'objet auquel vous souhaitez accéder. Dans ce scénario, l'objectif est d'accéder à l'objet **Account**.

7. Dans la liste **Schema**, sélectionnez **Repository** puis cliquez sur le bouton **[...]** pour sélectionner le schéma du **Repository** que vous souhaitez utiliser pour le composant. Si vous n'avez pas enregistré le schéma dans les métadonnées du **Repository**, sélectionnez **Built-in** dans la liste **Schema** et cliquez sur le bouton **[...]** à côté du champ **Edit schema** et paramétrez le schéma manuellement.

8. Dans le champ **Query Condition**, saisissez la requête que vous souhaitez appliquer. Pour ce scénario, l'objectif est de récupérer le ou les clients dont le nom est *sForce*., ainsi, saisissez la requête : `"name='sForce'"`.

9. Pour une requête plus avancée, cochez la case **Manual input of SOQL query** et saisissez-la manuellement.

Supprimer le contenu interrogé

1. Double-cliquez sur le composant **tSalesforceOutput** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés :

2. Dans le champ **Salesforce WebService URL**, laissez l'URL du service Web Salesforce par défaut ou saisissez l'URL à laquelle vous souhaitez accéder.

3. Dans les champs **Username** et **Password**, saisissez votre identifiant et votre mot de passe de connexion au service Web.

4. Saisissez, en millisecondes, le délai avant suspension dans le champ **Timeout (milliseconds)**. Dans ce scénario, utilisez la valeur par défaut.

5. Dans la liste **Action**, sélectionnez l'action que vous souhaitez effectuer. Ici, sélectionnez **Delete** pour supprimer le compte *sForce* sélectionné dans le composant précédent.

6. Dans la liste **Module**, sélectionnez l'objet auquel vous souhaitez accéder. Dans ce scénario, l'objectif est d'accéder à l'objet **Account**.

7. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

8. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

Consultez le contenu de l'objet **Account** et vérifiez que le compte *sForce* a été supprimé du serveur.

Scénario 2 : Rassembler des données en erreur tout en insérant des données dans un module sur Salesforce.com

Dans ce scénario, les données d'un fichier local sont insérées dans le module **AdditionalNumber**. Pendant ce temps, les données en erreur de ce fichier sont collectées via un lien **Row** > **Reject**.

Déposer et relier les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tFileInputDelimited**, **tSalesforceOutput** et deux **tLogRow**.

2. Renommez le **tFileInputDelimited** en *DataToInsert*, le **tSalesforceOutput** en *InsertToSalesforce* et les deux **tLogRow** en *DataInserted* et *DataRejected*, respectivement.

3. Reliez le **DataToInsert** au **InsertToSalesforce** à l'aide d'un lien **Row** > **Main**.

4. Reliez le **InsertToSalesforce** au **DataInserted** à l'aide d'un lien **Row** > **Main**.

5. Reliez le **InsertToSalesforce** au **DataRejected** à l'aide d'un lien **Row** > **Reject**.

Le lien **Reject** est disponible uniquement lorsque les cases **Extended Output** et **Die on error** sont décochées dans la vue **Advanced settings** du composant **tSalesforceOutput**.

Configurer la source des données

1. Double-cliquez sur le composant **DataToInsert** pour ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Dans la liste **Property Type**, sélectionnez **Built-In**.

Vous pouvez sélectionner **Repository** dans la liste **Property Type** pour renseigner automatiquement les propriétés si la métadonnée correspondante est stockée dans le **Repository**. Pour plus d'informations concernant la centralisation des métadonnées, consultez le *Guide utilisateur du Studio Talend*.

3. Dans le champ **File name/Stream**, saisissez le chemin d'accès à votre fichier source, ou parcourez votre système jusqu'à celui ci. Dans cet exemple, saisissez *E:/salesforceout.csv*.

4. Dans le champ **Header**, saisissez *1* pour récupérer le nom des colonnes. Laissez les autres paramètres tels qu'ils sont.

Configurer le module pour l'insertion de données

1. Double-cliquez sur le **InsertToSalesforce** pour ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Dans le champ **Username**, saisissez votre identifiant, par exemple, *cantoine@talend.com*.

3. Dans le champ **Password**, saisissez votre mot de passe, par exemple, *talendehmrEvHz2xZ8f2KlmTCymS0XU*.

4. Dans la liste **Action**, sélectionnez **insert**.

5. Dans la liste **Module**, sélectionnez **AdditionalNumber**.

   Les cases **Extended Output** et **Die on error** ont précédemment été décochées dans la vue **Advanced settings** afin de faire apparaître le lien **Reject**.

6. Laissez les autres champs tels qu'ils sont.

Configurer l'affichage des résultats et exécuter le Job

1. Double-cliquez sur le composant **DataInserted** pour ouvrir sa vue **Basic settings**.

2. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour une meilleure présentation des résultats.

3. Répétez l'opération pour le composant **DataRejected**.

4. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

5. Appuyez sur la touche **F6** pour exécuter le Job. Vous pouvez constater que les données en erreur (s'il y en a) sont affichées dans la vue **Run**.

```
|
tLogRow_2
|
|=-----------------+------------------+----------------------+--------------
---+-------------------------+------------+--------------------------------
----------------------------=|
|CallCenterId      |Name              |Description           |Phone
|errorCode                |errorFields |errorMessage
|
|=-----------------+------------------+----------------------+--------------
---+-------------------------+------------+--------------------------------
----------------------------=|
|03m70000000CaZBAA0|Talend Testing V19|Conduct testing on
TOS|0086-010-88888888|FIELD_INTEGRITY_EXCEPTION|CallCenterId|Call Center ID:
id value of incorrect type: 03m70000000CaZBAA0|
|04m70000000CaZBAA0|Talend Testing V19|Conduct testing on
TOS|0086-010-88888888|FIELD_INTEGRITY_EXCEPTION|CallCenterId|Call Center ID:
id value of incorrect type: 04m70000000CaZBAA0|
'-----------------+------------------+----------------------+--------------
---+-------------------------+------------+--------------------------------
----------------------------'
```

Comme affiché dans la capture d'écran, les données de deux champs *Call Center ID* sont incorrectes.

Scénario 3 : Insérer des ID d'un fichier Excel vers le Module Contact

Dans ce scénario, les AccountIDs d'un fichier Excel sont insérés dans le module **Contact** sur `www.salesforce.com` selon la correspondance des champs *LastName* et *Name*.

Déposer les composants

1. Glissez-déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tFileInputExcel**, un **tSalesforceIntput**, un **tMap** et un **tSalesforceOutput**.
2. Renommez le **tFileInputExcel** en **excel_source**, le **tSalesforceIntput** en **load_salesforce_data**, le **tMap** en **match_and_output** et le **tSalesforceOutput** en **insert_to_contact_module**.
3. Reliez les composants à l'aide de liens **Row** > **Main**.

Configurer les composants

Configurer le fichier Excel source

1. Double-cliquez sur le composant nommé **excel_source** pour ouvrir sa vue **Component**.

2. Cliquez sur le bouton **[...]** à côté du champ **File name/Stream** afin de sélectionner le fichier source.

   Il se présente comme suit :

AccountId	LastName	Name
001A000000Ho08zIAB	Gonzalez	Gonzalez Rose
001A000000Ho090IAB	Forbes	Forbes Sean
001A000000Ho096IAB	Rogers	Rogers Jack

3. Cochez la case **All sheets** pour récupérer toutes les données du fichier Excel.

4. Saisissez *1* dans le champ **Header** afin que la première ligne contienne le nom des colonnes.

5. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

6. Cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes, respectivement *AccountId*, *LastName* et *Name*.

7. Cliquez sur **OK** pour fermer l'éditeur. Laissez les autre paramètres tels qu'ils sont.

Configurer le module de destination et l'opération souhaitée

1. Double-cliquez sur le composant nommé **insert_to_contact_module** pour ouvrir sa vue **Component**.

2. Dans les champs **Username** et **Password**, saisissez vos informations de connexion.

3. Sélectionnez **insert** dans la liste **Action** et **Contact** dans la liste **Module**.

4. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

5. Cliquez sur le bouton pour copier toutes les colonnes de la table de sortie dans la table d'entrée.

6. Cliquez sur **OK** pour fermer l'éditeur.

Configurer la source de référence et établir le mapping

1. Double-cliquez sur **load_salesforce_data** pour ouvrir sa vue **Component**.

2. Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant et votre mot de passe.

3. Dans la liste **Module**, sélectionnez le module **Contact** afin de récupérer les données.

4. Dans le champ **Query Condition**, supprimez le texte.

5. Double-cliquez sur **match_and_output** afin d'ouvrir l'éditeur de mapping.

6. Sélectionnez les champs **LastName** et **Name** de la table *row1* et déposez-les sur les lignes du même nom, dans la table *row2*. Ainsi, les données du fichier Excel sont vérifiées par rapport à celles du même nom dans le module **Contact**.

7. Sélectionnez les champs **LastName** et **AccountID** de la table *row1* et déposez-les sur les lignes du même nom, dans la table *id*. Ainsi, les données qualifiées du fichier Excel sont passées aux champs de même nom dans la table *id*.

8. Cliquez sur **OK** pour fermer l'éditeur.

Exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** pour exécuter le Job.

```
Starting job salesforce at 14:23 02/08/2012.

[statistics] connecting to socket on port 3671
[statistics] connected
[statistics] disconnected
Job salesforce ended at 14:23 02/08/2012.
[exit code=0]
```

Scénario 4 : Effectuer un upsert du module Contact selon les relations de mapping avec les ID externes dans le module Account

Ce scénario effectue un upsert des enregistrements du module Contact selon le mapping (matching) par rapport aux ID externes dans le module Account :

Relier les composants

1. Déposez un **tSalesforceConnection**, deux **tFixedFlowInput**, deux **tSalesforceInput**, deux **tSalesforceOutput** et deux **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Renommez les deux composants **tFixedFlowInput** en **external ids to insert** et **emails to upsert**, les deux **tSalesforceInput** en **Contact (in)** et **Account (in)**, les deux **tSalesforceOutput** en **Contact (out)** et **Account (out)** et les deux **tLogRow** en **external ids inserted** et **emails upserted**.
3. Reliez le **tSalesforceConnection** au **external ids to insert** à l'aide d'un lien **OnSubjobOk**.
4. Reliez le **external ids to insert** au **Account (out)** à l'aide d'un lien **Row** > **Main**.
5. Reliez le **external ids to insert** au **Account (in)** à l'aide d'un lien **OnSubjobOk**.
6. Reliez le **Account (in)** au **external ids inserted** à l'aide d'un lien **Row** > **Main**.
7. Reliez le composant **Account (in)** au **emails to upsert** à l'aide d'un lien **OnSubjobOk**.
8. Reliez le **emails to upsert** au **Contact (out)** à l'aide d'un lien **Row** > **Main**.
9. Reliez le **emails to upsert** au **Contact (in)** à l'aide d'un lien **OnSubjobOk**.
10. Reliez le composant **Contact (in)** au **emails upserted** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tSalesforceConnection** pour ouvrir sa vue **Basic settings**.

Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant et votre mot de passe.

2. Double-cliquez sur le **external ids to insert** pour ouvrir sa vue **Basic settings**.

3. Cliquez sur le bouton **Edit schema** pour ouvrir l'éditeur de schéma.

Cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes. Nommez-les respectivement *Name*, *AccountID__c* et *AccountBizLicense__c*, toutes de type String. Vous pouvez remarquer que les colonnes *AccountID__c* et *AccountBizLicense__c* sont des champs personnalisés du module Account, avec l'attribut de l'ID externe.

Cliquez sur **OK** pour fermer l'éditeur.

Cochez la case **Use Inline Content (delimited file)** dans la zone **Mode** et saisissez les données ci-dessous dans le champ **Content** :

```
Google;US666;C.A.666
Talend;FR888;Paris888
```

4. Double-cliquez sur le composant **Account (out)** pour ouvrir sa vue **Basic settings**.

Cochez la case **Use an existing connection**.

Sélectionnez **insert** dans la liste **Action** et **Account** dans la liste **Module**.

5. Double-cliquez sur **Account (in)** pour ouvrir sa vue **Basic settings**.

Cochez la case **Use an existing connection**.

Sélectionnez **Query** dans la liste **Query mode** et **Account** dans la liste **Module**.

Dans le champ **Query Condition**, saisissez l'instruction de filtre : `"name like 'Tal%' OR name like 'Goo%'"`.

6. Cliquez sur le bouton **Edit schema** pour ouvrir l'éditeur de schéma.

Cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes. Nommez-les respectivement *Name*, *AccountID__c* et *AccountBizLicense__c*, toutes de type String.

Cliquez sur **OK** pour fermer l'éditeur.

7. Double-cliquez sur le composant **external ids inserted** pour ouvrir sa vue **Basic settings**.

Cochez la case **Table (print values in cells of a table)** pour un affichage optimal des résultats.

8. Double-cliquez sur **emails to upsert** pour ouvrir sa vue **Basic settings**.

9. Cliquez sur le bouton **Edit schema** pour ouvrir l'éditeur du schéma.

Cliquez quatre fois sur le bouton **[+]** pour ajouter quatre colonnes. Nommez-les *Email*, *AccountID*, *AccountBizLicense* et *LastName*, toutes de type String.

Cliquez sur **OK** pour fermer l'éditeur.

Cochez la case **Use Inline Content (delimited file)** dans la zone **Mode** et saisissez les données ci-dessous dans le champ **Content** :

```
andy@talend.com;Paris888;FR888;Andy
anderson@talend.com;C.A.666;US666;Anderson
```

10. Double-cliquez sur **Contact (out)** pour ouvrir sa vue **Basic settings**.

Cochez la case **Use an existing connection**.

Sélectionnez **upsert** dans la liste **Action**, **Email** dans la liste **Upsert Key Column** et **Contact** dans la liste **Module**.

Cliquez sur l'onglet **Advanced settings** afin de configurer le mapping relationnel :

Cliquez deux fois sur le bouton **[+]** pour ajouter deux lignes et sélectionnez *AccountBizLicense* et *AccountID* dans la liste, dans la colonne **Column name of Talend Schema**.

Saisissez le nom des champs de relations lookup dans la colonne **Lookup field name**, à savoir *Account* et *Account__r*.

Saisissez le nom du module dans la colonne **Module name**, à savoir **Account**.

Saisissez le nom des champs d'IDs externes dans la colonne **External id name**, à savoir *AccountBizLicense__c* et *AccountID__c*. Ces champs correspondent aux champs personnalisés du module **Account**, avec l'attribut de l'ID externe.

- **Column name of Talend Schema** fait référence aux champs du schéma du composant précédent le **tSalesforceOutput**. Ces colonnes sont faites pour mettre en correspondance les champs d'IDs externes spécifiés dans la colonne **External id name**. Ces champs sont ceux spécifiés dans la colonne **Module name** du module de lookup.
- **Lookup field name** fait référence aux champs de relations lookup du module sélectionné dans la liste **Module** de la vue **Basic settings**. Ils servent à établir les relations avec le module de lookup spécifié dans le champ **Module name**.

- Pour plus d'informations concernant la définition des champs de relation lookup et comment saisir le bon nom dans le champ **Lookup field name**, rendez-vous sur le site Web de Salesforce et lancez l'application Salesforce Data Loader.

11. Double-cliquez sur le **Contact (in)** pour ouvrir sa vue **Basic settings**.

Cochez la case **Use an existing connection**.

Sélectionnez **Query** dans la liste **Query mode** et **Contact** dans la liste **Module**.

Dans le champ **Query Condition**, saisissez l'instruction de filtre : `"Email like 'And%'"`.

12. Cliquez sur le bouton **Edit schema** pour ouvrir l'éditeur de schéma.

Cliquez deux fois sur le bouton **[+]** pour ajouter deux colonnes de type String. Nommez-les respectivement *LastName* et *Email*.

Cliquez sur **OK** pour fermer l'éditeur.

13. Double-cliquez sur le composant **emails upserted** pour ouvrir sa vue **Basic settings**.

Sélectionnez l'option **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** pour exécuter le Job.

```
.------+------------+--------------------.
|         external ids inserted           |
|=-----+------------+--------------------=|
|Name  |AccountID__c|AccountBizLicense__c|
|=-----+------------+--------------------=|
|Google|US666       |C.A.666             |
|Talend|FR888       |Paris888            |
'------+------------+--------------------'
.--------+------------------.
|      emails upserted      |
|=-------+------------------=|
|LastName|Email              ||
|=-------+------------------=|
|Anderson|anderson@talend.com|
|Andy    |andy@talend.com    |
'--------+------------------'
```

Comme affiché ci-dessus, les actions d'insertion et de mise à jour ont bien été effectuées.

tSalesforceOutputBulk

Les composants **tSalesforceOutputBulk** et **tSalesforceBulkExec** sont généralement utilisés ensemble pour générer en sortie le fichier défini et exécuter ensuite les actions souhaitées sur votre fichier, dans votre Salesforce.com. Cette exécution en deux étapes est unifiée dans le composant **tSalesforceOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Propriétés du tSalesforceOutputBulk

Famille de composant	Business/Cloud	
Fonction	Le composant **tSalesforceOutputBulk** génère des fichiers dans un format approprié au traitement de masse.	
Objectif	Ce composant prépare le fichier devant être traité par le **tSalesForceBulkExec** afin d'exécuter des actions sur Salesforce.com.	
Basic settings	*File Name*	Parcourez le répertoire où stocker les fichiers générés.
	Append	Cochez cette case afin d'écrire à la suite des données existantes. Sinon, les données seront écrasées.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
	Ignore NULL fields values	Cochez cette case pour ignorer les valeurs nulles en mode **Update** ou **Upsert**.
Advanced settings	*Relationship mapping for upsert* (pour upsert uniquement)	Cliquez sur le bouton **[+]** pour ajouter des lignes et spécifiez les champs d'ID externes du flux d'entrée, les champs de relations lookup dans le module d'upsert, le module de lookup ainsi que les champs d'ID externes dans le module de lookup. De plus, la case **Polymorphic** doit être cochée uniquement lorsque les champs polymorphiques sont utilisés pour le mapping de relations. Pour plus d'informations concernant les champs polymorphiques, recherchez *polymorphic* sur http://www.salesforce.com/us/developer/docs/api_asynch/ (en anglais).

		Column name of Talend schema : champ d'ID externe dans le flux d'entrée. **Lookup field name** : champ de relation lookup dans le module d'upsert. **External id name** : champ d'ID externe dans le module de lookup. **Polymorphic** : cochez cette case uniquement lorsque les champs polymorphiques sont utilisés pour le mapping de relations. **Module name** : nom du module de lookup. • **Column name of Talend schema** fait référence aux champs du schéma du composant précédent le **tSalesforceOutput**. Ces colonnes sont mises en correspondance par rapport aux champs d'IDs externes spécifiés dans la colonne **External id name**. Ces champs sont ceux spécifiés dans la colonne **Module name** du module de lookup. • **Lookup field name** fait référence aux champs de relations lookup du module sélectionné dans la liste **Module** de la vue **Basic settings**. Ils servent à établir les relations avec le module de lookup spécifié dans le champ **Module name**. • Pour plus d'informations concernant la définition des champs de relation lookup et comment saisir le bon nom dans le champ **Lookup field name**, rendez-vous sur le site Web de Salesforce et lancez l'application Salesforce Data Loader. • Cochez la case **Polymorphic** uniquement pour les champs polymorphiques. Vous pouvez obtenir une erreur si vous ne cochez pas cette case pour un champ polymorphique. Vous pouvez également obtenir une erreur si vous la sélectionnez pour un champ qui n'est pas polymorphique.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé avec le **tSalesforceBulkExec**. Ils permettent d'optimiser les performances d'alimentation ou de modification des informations sur Salesforce.com.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur	

	le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Insérer des données de masse transformées dans votre Salesforce.com

Ce scénario décrit un Job à six composants qui transforme des données *.csv* appropriées à un traitement de masse, les charge dans Salesforce.com et affiche ensuite les résultats de l'exécution du Job dans la console.

Ce Job est réalisé en deux étapes : préparer les données en les transformant, puis traiter les données transformées.

Avant de commencer ce scénario, vous devez préparer le fichier original comprenant les données devant être traitées par le Job. Dans cet exemple d'utilisation, le fichier est *sforcebulk.txt*, il contient des informations relatives aux clients.

Afin de créer et exécuter ce Job, procédez comme suit :

Configurer le Job

1. Glissez-déposez de la **Palette** dans l'espace de modélisation graphique les composants **tFileInputDelimited**, **tMap**, **tSalesforceOutputBulk**, **tSalesforceBulkExec** et deux **tLogRow**.

2. Utilisez un lien **Row** > **Main** afin de connecter le **tFileInputDelimited** au **tMap**, et le lien **Row** > **out1** du **tMap** au **tSalesforceOutputBulk**.

3. Utilisez un lien **Row** > **Main** et un lien **Row** > **Reject** pour connecter le **tSalesforceBulkExec** aux deux **tLogRow** respectivement.

4. Utilisez un lien **Trigger** > **OnSubjobOk** afin de connecter le composant **tFileInputDelimited** au **tSalesforceBulkExec**.

Configurer le composant d'entrée

1. Double-cliquez sur le **tFileInputDelimited** afin d'afficher sa vue **Component** et définir ses propriétés.

2. Dans la liste **Property Type**, sélectionnez **Repository** si vous avez déjà stocké votre connexion Salesforce sous le nœud **Metadata** dans la vue **Repository**. Les propriétés sont automatiquement remplies. Si vous n'avez pas défini le serveur de connexion localement dans le Repository, renseignez les informations manuellement, après avoir sélectionné **Built-in** dans la liste **Property Type**.

   Pour plus d'informations concernant la création des métadonnées de fichier délimité, consultez le *Guide utilisateur du Studio Talend*.

3. A côté du champ **File name/Stream**, cliquez sur le bouton ... afin de parcourir votre répertoire jusqu'à votre fichier, dans cet exemple *sforcebulk.txt*.

4. Dans la liste **Schema**, sélectionnez **Repository**, puis cliquez sur le bouton **[...]** afin d'ouvrir la boîte de dialogue dans laquelle vous pourrez sélectionner le schéma du Repository à utiliser pour ce composant. Si vous n'avez pas défini localement votre fichier dans les métadonnées, sélectionnez **Built-in** dans la liste **Schema**, puis cliquez sur le bouton **[...]** à côté du champ **Edit schema**, afin d'ouvrir une boîte de dialogue dans laquelle vous pourrez définir le schéma manuellement. Dans ce scénario, le schéma se compose de quatre colonnes : *Name*, *ParentId*, *Phone* et *Fax*.

Column	Key	Type	N...	Date Patt...	Length	Pre...	D...	Co...
Name		String	✓					
ParentId		String	✓					
Phone		String	✓					
Fax		String	✓					

5. Selon votre fichier d'entrée, paramétrez les autres champs, comme **Row Separator**, **Field Separator**.

Configurer le mapping

1. Double-cliquez sur le composant **tMap** afin d'ouvrir son éditeur et définir la transformation.

2. Glissez-déposez toutes les colonnes de la table d'entrée dans la table de sortie.

Ajoutez `.toUpperCase()` à la fin de la colonne *Name*.

3. Cliquez sur **OK** afin de valider la transformation.

Définir le chemin de sortie

1. Double-cliquez sur le **tSalesforceOutputBulk** afin d'afficher sa vue **Component** et définir ses propriétés.

2. Dans le champ **File Name**, parcourez votre répertoire jusqu'à l'emplacement où vous souhaitez stocker les données *.csv* générées pour un traitement de masse.

3. Cliquez sur le bouton **Sync columns** afin d'importer le schéma du composant précédent.

Configurer la connexion au serveur Salesforce

1. Double-cliquez sur le **tSalesforceBulkExec** afin d'afficher sa vue **Component** et définir ses propriétés.

2. Utilisez l'URL par défaut du service Web Salesforce, ou saisissez l'URL à laquelle vous souhaitez accéder.

3. Dans les champs **Username** et **Password**, saisissez votre identifiant et votre mot de passe de connexion au service Web.

4. Dans le champ **Bulk file path**, parcourez votre répertoire jusqu'à l'emplacement où est stocké le fichier *.csv* généré par le **tSalesforceOutputBulk**.

5. Dans la liste **Action**, sélectionnez l'action que vous voulez effectuer sur les données de masse. Dans cet exemple, sélectionnez **insert**.

6. Dans la liste **Module**, sélectionnez l'élément auquel vous souhaitez accéder, **Account** dans cet exemple.

7. Dans la liste **Schema**, sélectionnez **Repository** et cliquez sur le bouton **[...]** afin d'ouvrir une boîte de dialogue dans laquelle vous pourrez sélectionner le schéma du Repository à utiliser pour ce composant. Si vous n'avez pas défini votre schéma localement dans les métadonnées, sélectionnez **Built-in** dans la liste **Schema**, puis cliquez sur le bouton **[...]** à côté du champ **Edit schema**, afin d'ouvrir une boîte de dialogue dans laquelle vous pourrez définir votre schéma manuellement. Dans cet exemple, définissez-le comme le schéma précédent.

Configurer le composant de sortie

1. Double-cliquez sur le **tLogRow_1** afin d'afficher sa vue **Component** et définir ses propriétés.

2. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.

3. Sélectionnez le mode **Table** afin d'afficher les résultats d'exécution dans un tableau.

4. Répétez l'opération avec le **tLogRow_2**.

Exécuter le Job

1. Sauvegardez votre Job.

2. Appuyez sur **F6** pour l'exécuter.

   Dans la console de la vue **Run**, vous pouvez vérifier les résultats d'exécution.

```
[statistics] connected
.-----------------------------------+--------+--------------+--------------+-----------------+-----
|                                                   tLogRow_1
|=----------------------------------+--------+--------------+--------------+-----------------+-----
|Name                               |ParentId|Phone         |Fax           |salesforce_id    |sales
|=----------------------------------+--------+--------------+--------------+-----------------+-----
|BURLINGTON TEXTILES CORP OF AMERICA|null    |(336) 222-7000|(336) 222-8000|0018000000gfDyDAAU|true
|DICKENSON PLC                      |null    |(785) 241-6200|(785) 241-6201|0018000000gfDyEAAU|true
|GENEPOINT                          |null    |(650) 867-3450|(650) 867-9895|0018000000gfDyFAAU|true
'-----------------------------------+--------+--------------+--------------+-----------------+-----

.-------------------------+--------+--------------+--------------+-------------------------------
|                                                      tLogRow_2
|=------------------------+--------+--------------+--------------+-------------------------------
|Name                     |ParentId|Phone         |Fax           |error
|=------------------------+--------+--------------+--------------+-------------------------------
|EDGE COMMUNICATIONS      |wrong id|(512) 757-6000|(512) 757-9000|MALFORMED_ID:Parent Account ID: :
|GRAND HOTELS & RESORTS LTD|wrong id|(312) 596-1000|(312) 596-1500|MALFORMED_ID:Parent Account ID: :
'-------------------------+--------+--------------+--------------+-------------------------------

[statistics] disconnected
Job tSalesforcebulk ended at 18:17 13/04/2010. [exit code=0]
```

Dans le tableau du **tLogRow_1**, vous pouvez lire les données insérées dans votre Salesforce.com.

Dans le tableau du **tLogRow_2**, vous pouvez lire les données rejetées à cause d'une incompatibilité avec les éléments **Account** auxquels vous avez accédé.

Tous les noms de clients sont écrits en majuscules.

tSalesforceOutputBulkExec

Les composants **tSalesforceOutputBulk** et **tSalesforceBulkExec** sont généralement utilisés ensemble pour générer en sortie le fichier défini et exécuter ensuite les actions souhaitées sur votre fichier, dans votre Salesforce.com. Cette exécution en deux étapes est unifiée dans le composant **tSalesforceOutputBulkExec**. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Propriétés du tSalesforceOutputBulkExec

Famille de composant	Business/Cloud	
Fonction	Le composant **tSalesforceOutputBulkExec** exécute les actions définies sur les données *.csv* pour Salesforce.com.	
Objectif	En tant que composant dédié, le **tSalesforceOutpuBulkExec** permet d'optimiser les performances lors des opérations définies sur les données, dans votre Salesforce.com.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Login Type	Deux options sont disponibles : **Basic** : sélectionnez cette option afin de vous connecter à Salesforce.com en saisissant vos identifiants (nom d'utilisateur et mot de passe) dans le **tSalesforceConnection**. **OAuth2** : sélectionnez cette option pour accéder à Salesforce.com en saisissant votre Consumer key/Consumer Secret dans le **tSalesforceConnection**. Ainsi, vos identifiants ne sont pas visibles dans le **tSalesforceConnection**, mais un effort supplémentaire est requis : • Définissez une application connectée (Connected App) sur Salesforce.com pour obtenir votre Consumer key/Consumer Secret. Pour plus d'informations concernant les applications connectées, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais). • Lors de l'exécution d'un Job, la console du Studio affiche l'URL à traiter via OAuth. Pour plus d'informations, consultez *Scénario : Utiliser OAuth2 pour vous connecter à Salesforce.com*.
	Salesforce Webservice URL	Saisissez l'URL du service Web permettant de se connecter à la base de données Salesforce.
	Salesforce Version	Saisissez la version de Salesforce que vous utilisez.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.

	Consumer Key et *Consumer Secret*	Saisissez les détails de votre authentification OAuth. Ces informations sont disponibles dans la zone **OAuth Settings** de l'application connectée créée sur le site Salesforce.com. Pour saisir le Secret du consumer, cliquez sur le bouton **[...]** à côté du champ **Consumer Secret**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le Secret du consumer entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Pour comprendre ce qu'est une application connectée, consultez Connected Apps (en anglais). Pour plus d'informations concernant la création d'une application connectée, consultez Defining Remote Access Applications (en anglais).
	Callback Host et *Callback Port*	Saisissez l'URL de votre fonction de rappel pour l'authentification OAuth. Cette URL (contenant l'hôte et le port) est définie durant la création d'une application connectée et apparaît dans la zone **OAuth Settings** de l'application connectée.
	Token File	Saisissez le nom du jeton de fichier. Il stocke le jeton de rafraîchissement utiliser pour récupérer le jeton d'accès sans autorisation.
	Bulk file path	Chemin d'accès au répertoire où sont stockées les données de masse à traiter.
	Action	Vous pouvez effectuer l'une des opérations suivantes sur les données de l'objet Salesforce : **Insert** : insérer les données. **Update** : mettre les données à jour. **Upsert** : mettre à jour et insérer les données. **Delete** : supprimer les données.
	Upsert Key Column	Spécifiez la colonne de clé pour l'opération d'upsert. Disponible lorsque l'option **Upsert** et sélectionnée dans la liste **Action**.
	Module	Sélectionnez l'objet adéquat dans la liste. **Custom Object** : cette option fait apparaître le champ **Custom Module Name** dans lequel vous pouvez saisir l'objet auquel vous souhaitez vous connecter.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

Advanced settings	*Rows to commit*	Saisissez le nombre de lignes devant être traitées par lot de données.
	Bytes to commit	Saisissez le nombre d'octets par lot de données devant être traités.
	Concurrency mode	Mode de simultanéité pour le Job : **Parallel** : Traite les lots en mode parallèle. **Serial** : Traite les lots en mode séquentiel.
	Wait time for checking batch state(milliseconds)	Saisissez l'intervalle de temps entre les vérifications de traitement des lots de données d'un Job jusqu'à ce que tous les lots soient traités.
	Use Socks Proxy	Cochez cette case si vous souhaitez utiliser un proxy. Dans ce cas, vous devez renseigner les paramètres du proxy dans les champs qui apparaissent, une fois la case cochée.
	Ignore NULL fields values	Cochez cette case pour ignorer les valeurs nulles en mode **Update** ou **Upsert**.
	Relationship mapping for upsert (pour **upsert** uniquement)	Cliquez sur le bouton **[+]** pour ajouter des lignes et spécifiez les champs d'ID externes du flux d'entrée, les champs de relations lookup dans le module d'upsert, le module de lookup ainsi que les champs d'ID externes dans le module de lookup. De plus, la case **Polymorphic** doit être cochée uniquement lorsque les champs polymorphiques sont utilisés pour le mapping de relations. Pour plus d'informations concernant les champs polymorphiques, recherchez *polymorphic* sur http://www.salesforce.com/us/developer/docs/api_asynch/ (en anglais). **Column name of Talend schema** : champ d'ID externe dans le flux d'entrée. **Lookup field name** : champ de relation lookup dans le module d'upsert. **External id name** : champ d'ID externe dans le module de lookup. **Polymorphic** : cochez cette case uniquement lorsque les champs polymorphiques sont utilisés pour le mapping de relations. **Module name** : nom du module de lookup. • **Column name of Talend schema** fait référence aux champs du schéma du composant précédent le **tSalesforceOutput**. Ces colonnes sont mises en correspondance par rapport aux champs d'IDs externes spécifiés dans la colonne **External id name**. Ces champs sont ceux spécifiés dans la colonne **Module name** du module de lookup. • **Lookup field name** fait référence aux champs de relations lookup du module sélectionné dans la liste **Module** de la vue **Basic settings**. Ils servent à établir les relations avec le module de lookup spécifié dans le champ **Module name**. • Pour plus d'informations concernant la définition des champs de relation lookup et comment saisir le bon nom dans le champ **Lookup field name**, rendez-vous sur le site Web de Salesforce et lancez l'application Salesforce Data Loader. • Cochez la case **Polymorphic** uniquement pour les champs polymorphiques. Vous pouvez obtenir une erreur si vous ne cochez pas cette case pour un champ polymorphique. Vous pouvez également

		obtenir une erreur si vous la sélectionnez pour un champ qui n'est pas polymorphique.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.	
Limitation	Les données de masse à traiter dans Salesforce.com doivent être au format *.csv*	

Scénario : Insérer des données de masse dans votre Salesforce.com

Ce scénario décrit un Job à quatre composants qui charge des données en masse dans Salesforce.com, exécute les actions définies, et affiche les résultats d'exécution du Job dans la console.

Avant de réaliser ce Job, vous devez préparer votre fichier d'entrée, contenant les données devant être traitées par le Job. Dans cet exemple, le fichier *sforcebulk.txt*, contient des informations relatives aux clients.

Afin de créer et exécuter le Job, procédez comme suit :

Configurer le Job

1. Glissez-déposez de la **Palette** dans l'espace de modélisation graphique un composant **tFileInputDelimited**, un **tSalesforceOutputBulkExec**, et deux **tLogRow**.
2. Utilisez un lien **Row** > **Main** pour connecter le **tFileInputDelimited** au **tSalesforceOutputBulkExec**.
3. Utilisez un lien **Row** > **Main** et un lien **Row** > **Reject** afin de connecter le **tSalesforceOutputBulkExec** aux deux **tLogRow** respectifs.

Configurer les données d'entrée

1. Double-cliquez sur le **tFileInputDelimited** afin d'afficher sa vue **Component** et définir ses propriétés.

2. Dans la liste **Property Type**, sélectionnez **Repository** si vous avez déjà stocké votre connexion au serveur Salesforce dans le nœud **Metadata** de la vue **Repository**. Les propriétés sont automatiquement renseignées. Si vous n'avez pas défini la connexion au serveur localement dans le Repository, remplissez les champs manuellement, après avoir sélectionné **Built-in** dans la liste **Property Type**.

   Pour plus d'informations concernant la création des métadonnées de fichier délimité, consultez le *Guide utilisateur du Studio Talend*.

3. A côté du champ **File name/Stream**, cliquez sur le bouton afin de sélectionner votre fichier d'entrée, par exemple *sforcebulk.txt*.

4. Dans la liste **Schema**, sélectionnez **Repository**, puis cliquez sur le bouton **[...]** pour ouvrir une boîte de dialogue dans laquelle vous pourrez sélectionner le schéma à utiliser pour ce composant. Si vous n'avez pas défini votre schéma localement dans les métadonnées, sélectionnez **Built-in** dans la liste **Schema**, puis cliquez sur le bouton **[...]** à côté du champ **Edit schema**, afin d'ouvrir une boîte de dialogue où vous pourrez définir le schéma manuellement. dans ce scénario, le schéma se compose de quatre colonnes : *Name*, *ParentId*, *Phone* et *Fax*.

Column	Key	Type	N...	Date Patt...	Length	Pre...	D...	Co...
Name		String	✓					
ParentId		String	✓					
Phone		String	✓					
Fax		String	✓					

5. Configurez les autres propriétés selon votre fichier d'entrée, comme **Row Separator**, **Field Separator**...

Configurer la connexion au serveur Salesforce

1. Double-cliquez sur le composant **tSalesforceOutputBulkExec** afin d'afficher sa vue **Component** et définir ses propriétés.

2. Dans le champ **Salesforce WebService URL**, utilisez l'URL par défaut du service Web Salesforce, ou saisissez l'URL à laquelle vous souhaitez accéder.

3. Dans les champs **Username** et **Password**, saisissez votre identifiant et votre mot de passe de connexion au service Web.

4. Dans le champ **Bulk file path**, parcourez votre répertoire jusqu'à l'emplacement où vous stockez les données *.csv* devant être traitées.

   Le fichier de masse à traiter doit être au format *.csv*.

5. Dans la liste **Action**, sélectionnez l'action à effectuer sur les données de masse, **insert** dans cet exemple.

6. Dans la liste **Module**, sélectionnez l'élément auquel vous souhaitez accéder, **Account** dans cet exemple.

7. Dans la liste **Schema**, sélectionnez **Repository**, puis cliquez sur le bouton **[...]** afin d'ouvrir une boîte de dialogue dans laquelle vous pourrez sélectionner le schéma du Repository à utiliser pour ce composant. Si vous n'avez pas défini votre schéma localement dans les métadonnées, sélectionnez **Built-in** dans la liste **Schema**, puis cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir une boîte de dialogue où vous pourrez définir manuellement le schéma. Dans cet exemple, définissez le schéma comme précédemment.

Exécuter le Job

1. Double-cliquez sur le **tLogRow_1** afin d'afficher sa vue **Component** et définir ses propriétés.

2. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.

3. Sélectionnez le mode **Table** afin d'afficher les résultats d'exécution dans un tableau.

4. Répétez l'opération pour le **tLogRow_2**.

5. Sauvegardez votre Job et appuyez sur **F6** afin de l'exécuter.

Dans la console de la vue **Run**, vous pouvez vérifier les résultats d'exécution.

```
[statistics] connected
.-----------------------------------+--------+--------------+--------------+------------------+-----
|                                                tLogRow_1
|=----------------------------------+--------+--------------+--------------+------------------+-----
|Name                               |ParentId|Phone         |Fax           |salesforce_id     |sales
|=----------------------------------+--------+--------------+--------------+------------------+-----
|Burlington Textiles Corp of America|null    |(336) 222-7000|(336) 222-8000|0018000000gfDyDAAU|true
|Dickenson plc                      |null    |(785) 241-6200|(785) 241-6201|0018000000gfDyEAAU|true
|GenePoint                          |null    |(650) 867-3450|(650) 867-9895|0018000000gfDyFAAU|true
'-----------------------------------+--------+--------------+--------------+------------------+-----

.-------------------------+--------+--------------+--------------+-----------------------------
|                                                        tLogRow_2
|=------------------------+--------+--------------+--------------+-----------------------------
|Name                     |ParentId|Phone         |Fax           |error
|=------------------------+--------+--------------+--------------+-----------------------------
|Edge Communications      |wrong id|(512) 757-6000|(512) 757-9000|MALFORMED_ID:Parent Account ID: :
|Grand Hotels & Resorts Ltd|wrong id|(312) 596-1000|(312) 596-1500|MALFORMED_ID:Parent Account ID: :
'-------------------------+--------+--------------+--------------+-----------------------------
```

Dans le tableau du **tLogRow_1**, vous pouvez lire les données insérées dans Salesforce.com.

Dans le tableau du **tLogRow_2**, vous pouvez lire les données rejetées à cause d'une incompatibilité avec les éléments **Account** auxquels vous avez accédé.

Si vous souhaitez transformer les données d'entrée avant de les charger, vous devez utiliser un **tSalesforceOutputBulk** et un **tSalesforceBulkExec** pour réaliser le Job complet. Pour plus d'informations à propos de l'utilisation de ces deux composants, consultez *Scénario : Insérer des données de masse transformées dans votre Salesforce.com*.

tSalesforceWaveBulkExec

Propriétés du tSalesforceWaveBulkExec

Famille de composant	Business/Salesforce	
Fonction	Le **tSalesforceWaveBulkExec** charge des données dans Salesforce Analytics Cloud à partir d'un fichier CSV local.	
Objectif	Ce composant permet de charger des données dans Salesforce Analytics Cloud à partir d'un fichier local.	
Basic settings	*Property Type*	Peut être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	User Name et *Password*	Saisissez les détails d'authentification du service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	End Point	Saisissez l'URL du WebService nécessaire pour se connecter à Salesforce, *https://login.salesforce.com/services/Soap/u/32.0* par exemple. Notez que la version dans l'URL doit être 32.0 ou ultérieure.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. • Pour les champs de type numérique (par exemple *byte*, *short*, *int*, *long*, *float*, *double* et *BigDecimal*), vous devez spécifier leur valeurs de longueur et de précision. dans l'éditeur de schéma. Les valeurs par défaut de longueur et précision sont respectivement *10* et *2*, mais vous pouvez spécifiez des valeurs personnalisées dans l'éditeur du schéma. • Pour les champs de type date, vous devez spécifier le format de la date dans l'éditeur de schéma. Pour plus d'informations concernant les formats de date supportés, consultez Analytics Cloud External Data Format Reference (en anglais). Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. La liste **Schema** et le bouton **Editor schema** ne sont pas disponibles pas si la case **Custom JSON Metadata** est cochée.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Operation	Sélectionnez l'opération à effectuer sur l'ensemble de données : • **Append** : Ajoute toutes les données à l'ensemble de données. Crée un ensemble de données s'il n'existe pas. • **Upsert** : Insère ou met à jour des lignes dans l'ensemble de données. Crée un ensemble de données s'il n'existe pas. • **Overwrite** : Crée un nouvel ensemble de données à partir des données fournies et remplace l'ensemble de données s'il existe déjà. • **Delete** : Supprime les lignes de l'ensemble de données. • Un fichier de métadonnées JSON est requis pour les opérations **Append**, **Upsert** et **Delete**. • Les données et les métadonnées des opérations **Append** et **Upsert** doivent correspondre à l'ensemble de données sur lequel l'opération est effectuée. Toutes les colonnes, les dimensions et les mesures doivent correspondre parfaitement. • L'opération **Append** n'est pas autorisée si vous définissez n'importe quelle colonne comme étant la clé primaire. • Vous devez spécifier une (et seulement une) colonne en tant que clé primaire sur laquelle l'opération **Upsert** ou **Delete** est basée. Vous pouvez le faire en cliquant sur **Edit schema** et en cochant la case située à côté de la colonne que vous souhaitez définir en tant que clé primaire. • Les métadonnées pour l'opération **Delete** doivent être un sous-ensemble des colonnes de l'ensemble de données.
	Name	Saisissez le nom de l'ensemble de données dans lequel charger les données.
	CSV File	Spécifiez le chemin vers le fichier CSV à charger.
Advanced settings	*Encoding*	Sélectionnez le type d'encodage utilisé pour générer le fichier de métadonnées JSON qui décrit la structure des données à charger.
SalesforceWave Metadata Descriptor	*Auto Generate JSON Metadata Description*	Cochez cette case pour générer automatiquement la description des métadonnées JSON.
	Header	Spécifiez le nombre de lignes à ignorer dans le fichier CSV. Ce champ est disponible uniquement lorsque la case **Auto Generate JSON Metadata Description** n'est pas cochée.
	Unique API Name	Spécifiez le nom unique de l'API pour l'objet dans la description des métadonnées JSON. Ce champ est disponible uniquement lorsque la case **Auto Generate JSON Metadata Description** n'est pas cochée.
	Label	Spécifiez le nom à afficher pour l'objet dans la description des métadonnées JSON. Ce champ est disponible uniquement lorsque la case **Auto Generate JSON Metadata Description** n'est pas cochée.
	Fully Qualified Name	Spécifiez le chemin complet qui identifie de manière unique l'enregistrement dans la description des métadonnées JSON. Ce champ est disponible uniquement lorsque la case **Auto Generate JSON Metadata Description** n'est pas cochée.
	Custom JSON Meatadata	Cochez cette case pour utiliser un fichier de métadonnées JSON personnalisé. Cette case est disponible uniquement lorsque la case **Auto Generate JSON Metadata Description** n'est pas cochée.

	JSON Meatadata	Spécifiez le chemin vers le fichier de métadonnées JSON personnalisé. Ce champ est disponible uniquement lorsque la case **Custom JSON Metadata** est cochée.
	Generate JSON in File	Cochez cette case pour écrire la description des métadonnées JSON dans un fichier local. Cette case est disponible uniquement lorsque la case **Custom JSON Metadata** est cochée.
	Folder	Spécifiez le répertoire dans lequel vous souhaitez stocker le fichier de métadonnées JSON généré. Ce champ est disponible uniquement lorsque la case **Generate JSON in File** est cochée.
	tStatCatcher Statistics	Cochez cette case pour récupérer les métadonnées relatives à l'exécution du Job, au niveau du Job et au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Insérer des données de masse transformées dans votre Salesforce.com*.

tSalesforceWaveOutputBulkExec

Propriétés du tSalesforceWaveOutputBulkExec

Famille de composant	Business/Salesforce	
Fonction	Le composant **tSalesforceWaveOutputBulkExec** reçoit des données depuis le composant précédent, génère un fichier CSV local, puis charge les données de ce fichier sur Salesforce Analytics Cloud.	
Objectif	En tant que composant dédié, il permet de gagner en performance durant les opérations de données sur Salesforce Analytics Cloud.	
Basic settings	*Property Type*	Peut être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	User Name et *Password*	Saisissez les détails d'authentification du service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	End Point	Saisissez l'URL du WebService nécessaire pour se connecter à Salesforce, *https://login.salesforce.com/services/Soap/u/32.0* par exemple. Notez que la version dans l'URL doit être 32.0 ou ultérieure.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. • Pour les champs de type numérique (par exemple *byte*, *short*, *int*, *long*, *float*, *double* et *BigDecimal*), vous devez spécifier leur valeurs de longueur et de précision. dans l'éditeur de schéma. Les valeurs par défaut de longueur et précision sont respectivement *10* et *2*, mais vous pouvez spécifiez des valeurs personnalisées dans l'éditeur du schéma. • Pour les champs de type date, vous devez spécifier le format de la date dans l'éditeur de schéma. Pour plus d'informations concernant les formats de date supportés, consultez Analytics Cloud External Data Format Reference (en anglais). Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. La liste **Schema** et le bouton **Editor schema** ne sont pas disponibles lorsque la case **Custom JSON Metadata** est cochée.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
	Operation	Sélectionnez l'opération à effectuer sur l'ensemble de données : • **Append** : Ajoute toutes les données à l'ensemble de données. Crée un ensemble de données s'il n'existe pas. • **Upsert** : Insère ou met à jour des lignes dans l'ensemble de données. Crée un ensemble de données s'il n'existe pas. • **Overwrite** : Crée un nouvel ensemble de données à partir des données fournies et remplace l'ensemble de données s'il existe déjà. • **Delete** : Supprime les lignes de l'ensemble de données. • Un fichier de métadonnées JSON est requis pour les opérations **Append**, **Upsert** et **Delete**. • Les données et les métadonnées des opérations **Append** et **Upsert** doivent correspondre à l'ensemble de données sur lequel l'opération est effectuée. Toutes les colonnes, les dimensions et les mesures doivent correspondre parfaitement. • L'opération **Append** n'est pas autorisée si vous définissez n'importe quelle colonne comme étant la clé primaire. • Vous devez spécifier une (et seulement une) colonne en tant que clé primaire sur laquelle l'opération **Upsert** ou **Delete** est basée. Vous pouvez le faire en cliquant sur **Edit schema** et en cochant la case située à côté de la colonne que vous souhaitez définir en tant que clé primaire. • Les métadonnées pour l'opération **Delete** doivent être un sous-ensemble des colonnes de l'ensemble de données.
	Name	Saisissez le nom de l'ensemble de données dans lequel charger les données.
	Generated CSV File	Spécifiez le chemin vers le fichier CSV à générer.
	Append	Cochez cette case pour écrire à la suite les données dans le fichier local spécifié s'il existe au lieu de l'écraser.
	Include Header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
Advanced settings	*Encoding*	Sélectionnez le type d'encodage utilisé pour générer le fichier de métadonnées JSON qui décrit la structure des données à charger.
	Create directory if not exists	Cochez cette case pour créer, s'il n'existe pas, le répertoire spécifié dans le champ **Generated CSV File**.
	Custom the flush buffer size	Cochez cette case pour spécifier le nombre de lignes à écrire avant de vider la mémoire tampon.
	Row number	Spécifiez le nombre de lignes à écrire avant de vider la mémoire tampon. Ce champ est disponible uniquement lorsque la case **Custom the flush buffer size** est cochée.
	Don't generate empty file	Cochez cette case si vous ne souhaitez pas générer de fichier vide.
SalesforceWave Metadata Descriptor	*Auto Generate JSON Metadata Description*	Cochez cette case pour générer automatiquement la description des métadonnées JSON.
	Header	Spécifiez le nombre de lignes à ignorer dans le fichier CSV. Ce champ est disponible uniquement lorsque la case **Auto Generate JSON Metadata Description** n'est pas cochée.
	Unique API Name	Spécifiez le nom unique de l'API pour l'objet dans la description des métadonnées JSON.

		Ce champ est disponible uniquement lorsque la case **Auto Generate JSON Metadata Description** n'est pas cochée.
	Label	Spécifiez le nom à afficher pour l'objet dans la description des métadonnées JSON. Ce champ est disponible uniquement lorsque la case **Auto Generate JSON Metadata Description** n'est pas cochée.
	Fully Qualified Name	Spécifiez le chemin complet qui identifie de manière unique l'enregistrement dans la description des métadonnées JSON. Ce champ est disponible uniquement lorsque la case **Auto Generate JSON Metadata Description** n'est pas cochée.
	Custom JSON Metadata	Cochez cette case pour utiliser un fichier de métadonnées JSON personnalisé. Cette case s'affiche uniquement lorsque la case **Auto Generate JSON Metadata Description** n'est pas cochée.
	JSON Metadata	Spécifiez le chemin vers le fichier de métadonnées JSON personnalisé. Ce champ est disponible uniquement lorsque la case **Custom JSON Metadata** est cochée.
	Generate JSON in File	Cochez cette case pour écrire la description des métadonnées JSON dans un fichier local. Cette case est disponible uniquement lorsque la case **Custom JSON Metadata** est cochée.
	Folder	Spécifiez le répertoire dans lequel vous souhaitez stocker le fichier de métadonnées JSON généré. Ce champ est disponible uniquement lorsque la case **Generate JSON in File** est cochée.
	tStatCatcher Statistics	Cochez cette case pour récupérer les métadonnées relatives à l'exécution du Job, au niveau du Job et au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger sur Salesforce Analytics Cloud.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Insérer des données de masse dans votre Salesforce.com*.

tSAPBWInput

Propriétés du tSAPBWInput

Famille de composant	Business	
Fonction	Le composant **tSAPBWInput** lit des données dans une base de données SAP BW à l'aide d'une connexion API JDBC et extrait des champs selon une requête SQL.	
Objectif	Ce composant exécute une requête SQL dans un ordre strictement défini devant correspondre à la définition de votre schéma. Il passe ensuite la liste des champs au composant suivant via un lien **Main** > **Row**.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	JDBC URL	Saisissez l'URL JDBC dans le champ associé pour vous connecter à la base de données. Par exemple, saisissez : *jdbc:jdbc4olap://adresse IP/nom de la base de données* pour vous connecter à une base de données SAP BW.
	Username	Saisissez l'identifiant d'authentification à la base de données.
	Password	Saisissez le mot de passe d'authentification à la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table Name	Saisissez le nom de la table.
	Query Type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.

		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Query	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Trim all the String/Char columns*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées. Décochez la case **Trim all the String/Char columns** pour activer la case **Trim columns**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données SAP BW via une connexion JDBC.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Lire des données d'une base de données SAP BW

Ce scénario décrit un Job comprenant deux composants, lisant des données d'une base de données SAP BW. Les données récupérées sont affichées dans la console.

Avant de configurer votre Job, assurez-vous que les prérequis suivants sont respectés :

1. Vous devez copier les fichiers .jar suivants composant le pilote jdbc4olap dans le chemin d'accès à votre classe :

- activation.jar

- commons-codec.jar

- jdbc4olap.jar

- saaj-api.jar

- saaj-impl.jar

2. Assurez-vous d'avoir la dernière version du pilote jdbc4olap. Vous pouvez télécharger la dernière version du pilote jdbc4olap sur jdbc4olap download section (en anglais). Pour plus d'informations concernant l'utilisation du pilote jdbc4olap, consultez le Guide jdbc4olap User Guide (en anglais).

Dans ce scénario, les quatre opérations suivantes sont nécessaires :

1. Configurez le Job.

2. Configurez la connexion JDBC au serveur SAP BW.

3. Configurez une requête.

4. Affichez les données récupérées dans la console.

Configurer le Job

1. Déposez un composant **tSAPBWInput** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.

2. Connectez le **tSAPBWInput** au **tLogRow** à l'aide d'un lien **Main** > **Row**.

Configurer la connexion JDBC au serveur SAP BW

1. Double-cliquez sur le composant **tSAPBWInput** pour ouvrir sa vue **Basic settings** et configurer ses propriétés.

2. Dans le champ **JDBC URL**, saisissez l'URL de votre serveur jdbc4olap.

   Notez que l'URL ci-dessus est uniquement affichée dans un but de démonstration.

3. Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant et votre mot de passe d'authentification à la base de données.

4. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir le schéma à utiliser.

5. Cliquez sur le bouton **[+]** pour ajouter de nouvelles colonnes au schéma et définissez le type de données pour chaque colonne et cliquez sur **OK** afin de sauvegarder vos paramètres.

Configurer une requête

1. Dans la vue **Basic settings** du composant **tSAPBWInput**, dans le champ **Table Name**, saisissez le nom de la table. Dans ce scénario, le nom de la table *"Measures"* est uniquement pour démonstration.

2. Dans le champ **Query**, saisissez le script de la requête. Dans cet exemple, saisissez :

```
"SELECT
T1.\"[0D_CO_CODE].[LEVEL01]\" AS company,
T0.\"[Measures].[D68EEPGGHUMSZ92PIJARDZ0KA]\" AS amount
FROM
\"0D_DECU\".\"0D_DECU/PRE_QRY4\".\"[Measures]\" T0,
```

```
\"0D_DECU\".\"0D_DECU/PRE_QRY4\".\"[0D_CO_CODE]\" T1 "
```

Les requêtes SQL supportées étant limitées, les scripts de requêtes à utiliser doivent être basés sur la grammaire définie dans le pilote jdbc4olap. Pour plus d'informations concernant cette grammaire, consultez le guide jdbc4olap User Guide (en anglais).

Afficher les données récupérées dans la console

1. Double-cliquez sur le composant **tLogRow** pour ouvrir sa vue **Basic settings** et configurer ses propriétés.

2. Cliquez sur le bouton **Sync columns** pour récupérer le schéma défini dans le composant précédent.
3. Sélectionnez **Table** dans la zone **Mode**.
4. Appuyez sur **Ctrl**+**S** pour sauvegarder votre Job et appuyez sur **F6** pour l'exécuter.

```
.-------------------+---------.
|          tLogRow_1          |
|=------------------+--------=|
|Country            |Amount   |
|=------------------+--------=|
|IDES               |976190.0 |
|Century Inc.       |321678.0 |
|ACE Technology     |1432954.0|
|Haitec Enterprise  |221580.0 |
|Cannon Electronics|552852.0 |
|#                  |0.0      |
'-------------------+---------'
```

Les données de la table *"Measure"* sont récupérées et affichées dans la console.

tSAPCommit

Propriétés du tSAPCommit

Famille de composant	Business/SAP	
Fonction	Valide les données traitées dans un Job, via un serveur connecté.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.	
Basic settings	*SAPConnection Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tSAPConnection** dans la liste.
	Release Connection	Cette case est sélectionnée par défaut. Elle vous permet de fermer une connexion à une base de données, une fois le commit effectué; Décochez cette case afin de continuer à utiliser la connexion sélectionnée, une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien **Row>Main** pour relier le **tSAPCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Release connection**, ou votre connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion SAP parmi celles prévues dans votre Job. Lorsqu'un paramètre dynamique est défini, l'option **SAPConnection Component List** correspondante est désactivée dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants SAP et notamment avec les composants **tSAPConnection** et **tSAPRollback**.	

Scénario associé

Pour un scénario associé au **tSAPCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tSAPConnection

Propriétés du tSAPConnection

Famille de composant	Business	
Fonction	Le composant **tSAPConnection** ouvre une connexion vers un système SAP afin d'effectuer une transaction.	
Objectif	Ce composant permet de commiter les données d'un Job entier en une seule transaction vers un système SAP.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	SAP JCO Version	Sélectionnez la version de SAP Java Connector dans la liste déroulante, **talend sap api**, **sap jco 2.*** ou **sap jco 3.***. Notez que la connexion SAP ouverte peut être réutilisée par le **tSAPBapi** et le **tSAPTableInput** uniquement si l'option **talend sap api**est sélectionnée.
	Connection configuration	**Client type** : Saisissez votre code de connexion SAP habituel. **Userid :** Saisissez l'identifiant de connexion utilisateur. **Password** : Saisissez le mot de passe. **Language** : Spécifiez la langue. **Host name** : Saisissez l'adresse IP du système SAP. **System number :** Saisissez le numéro du système. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Specify a shared destination	Cochez cette case et spécifiez la destination partagée définie du côté *Talend Runtime* dans le champ **Shared destination name**. Le champ **Shared destination name** permet aux composants tSAP* d'utiliser une destination partagée dans *Talend Runtime*, lorsqu'ils sont exécutés comme partie d'une Route ESB ou d'un service de données ESB dans l'environnement OSGi du conteneur de *Talend Runtime*. Cette option est disponible uniquement lorsque vous sélectionnez **sap jco 3.*** dans la liste déroulante **SAP JCO Version**. En plus de spécifier la destination partagée, dans le champ **Shared destination name** du composant, vous devez également configurer la source des données SAP (DataSource) pour *Talend Runtime*. La procédure pour définir la source des données SAP et pour la configurer sont décrites dans le fichier *lisezmoi.txt* fourni dans le package de Talend ESB, dans le dossier */add-ons/datasource/sap*.

		Si vous utilisez la configuration propre du composant, votre connexion SAP se ferme à la fin de l'exécution du composant. Pour éviter cela, utilisez une destination partagée ayant un nom spécifié.
	SAP Properties	Si vous souhaitez utiliser une configuration personnalisée pour le système SAP à utiliser, renseignez cette table avec la ou les propriété(s) à personnaliser. Par exemple, • si vous devez établir une connexion Unicode entre le système SAP et le *Studio Talend*, ajoutez la propriété *jco.server.unicode* et configurez sa valeur à *1*. • si vous devez configurer une connexion SAP avec SNC (Secure Network Communications), ajoutez les paramètres SNC et configurez leur valeur dans cette table. Pour plus d'informations concernant la configuration des paramètres SCN, consultez http://help.sap.com/saphelp_erp60_sp/helpdata/en/ce/1dfd3d4aefd95ee10000000a114084/content.htm et https://help.sap.com/javadocs/NW04/current/jc/com/sap/mw/jco/JCO.html (en anglais).
Advanced settings	*ftp and http based programs*	Pour invoquer du serveur SAP une fonction nécessitant le téléchargement d'un document, cochez cette case et assurez-vous que SAPGUI a bien été installé avec votre système SAP. Si la case est cochée mais que SAPGUI n'est pas installé, des erreurs vont subvenir. *Cette case est indisponible si vous cochez la case **Use an existing connection** dans l'onglet **Basic settings**.*
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec les composants SAP, notamment les composants **tSAPCommit** et **tSAPRollback**.	

Scénario associé

Pour un scénario associé, consultez *Scénario 1 : Récupérer les métadonnées d'un système SAP*.

tSAPInput

Propriétés du tSAPInput

Famille de composant	Business	Ce composant est déprécié.
Fonction	Le composant **tSAPInput** se connecte à un système SAP via l'adresse IP du système.	
Objectif	Il permet d'extraire des données d'un système SAP à n'importe quel niveau en faisant appel à des fonctions RFC ou BAPI.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant connexion de base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	SAP JCO Version	Sélectionnez la version de SAP Java Connector dans la liste déroulante.
	Connection configuration	**Client** : Saisissez votre code de connexion SAP habituel. **Userid :** Saisissez l'identifiant de connexion utilisateur. **Password** : Saisissez le mot de passe. **Language** : Spécifiez la langue. **Host name** : Saisissez l'adresse IP du système SAP. **System number :** Saisissez le numéro du système. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	FunName	Saisissez le nom de la fonction que vous souhaitez utiliser pour récupérer les données.
	Initialize input	Configurez les paramètres d'entrée. **Parameter Value** : Saisissez entre guillemets les valeurs correspondant au paramètre défini dans la colonne **Parameter Name**. **Type** : Sélectionnez le type d'entrée à récupérer.

		Table Name (Structure Name) : Saisissez entre guillemets le nom de la table. **Parameter Name** : Saisissez entre guillemets le nom du champ correspondant à la table définie dans la colonne **Table Name**. Lorsque plusieurs valeurs de paramètres utilisent le même nom de paramètre, vous devez saisir ces valeurs dans une seule ligne et les séparer par des virgules.
	Outputs	Configurez les paramètres des schémas à transmettre en sortie. **Schema** : Saisissez le nom du schéma à transmettre en sortie. **Type (for iterate)** : Sélectionnez le type de sortie que vous souhaitez obtenir. **Table Name (Structure Name)** : Saisissez entre guillemets le nom de la table. **Mapping** : Saisissez entre guillemets le nom des champs dont vous souhaitez récupérer les données. Vous pouvez définir autant de liens de sortie **Main** pour écrire des données que de schémas ajoutés au tableau **Outputs**. Ainsi, les données peuvent être groupées dans différents fichiers.
	Specify a shared destination	Cochez cette case et spécifiez la destination partagée définie du côté *Talend Runtime* dans le champ **Shared destination name**. Le champ **Shared destination name** permet aux composants tSAP* d'utiliser une destination partagée dans *Talend Runtime*, lorsqu'ils sont exécutés comme partie d'une Route ESB ou d'un service de données ESB dans l'environnement OSGi du conteneur de *Talend Runtime*. Cette option est disponible uniquement lorsque vous sélectionnez **sap jco 3.*** dans la liste déroulante **SAP JCO Version** et que vous ne cochez pas la case **Use an existing connection**. En plus de spécifier la destination partagée, dans le champ **Shared destination name** du composant, vous devez également configurer la source des données SAP (DataSource) pour *Talend Runtime*. La procédure pour définir la source des données SAP et pour la configurer sont décrites dans le fichier *lisezmoi.txt* fourni dans le package de Talend ESB, dans le dossier */add-ons/datasource/sap*. *Si vous utilisez la configuration propre du composant, votre connexion SAP se ferme à la fin de l'exécution du composant. Pour éviter cela, utilisez une destination partagée ayant un nom spécifié.*
Advanced settings	*SAP Properties*	Si vous souhaitez utiliser une configuration personnalisée pour le système SAP à utiliser, renseignez cette table avec la ou les propriété(s) à personnaliser. Par exemple, • si vous devez établir une connexion Unicode entre le système SAP et le *Studio Talend*, ajoutez la propriété *jco.server.unicode* et configurez sa valeur à *1*. • si vous devez configurer une connexion SAP avec SNC (Secure Network Communications), ajoutez les paramètres SNC et configurez leur valeur dans cette table. Pour plus d'informations concernant la configuration des paramètres SCN, consultez http://help.sap.com/saphelp_erp60_sp/helpdata/en/ce/1dfd3d4aefd95ee10000000a114084/content.htm et https://help.sap.com/javadocs/NW04/current/jc/com/sap/mw/jco/JCO.html (en anglais).

	ftp and http based programs	Pour invoquer du serveur SAP une fonction nécessitant le téléchargement d'un document, cochez cette case et assurez-vous que SAPGUI a bien été installé avec votre système SAP. Si la case est cochée mais que SAPGUI n'est pas installé, des erreurs vont subvenir. *Cette case est indisponible si vous cochez la case **Use an existing connection** dans l'onglet **Basic settings**.*
	Release Connection	Décochez cette case afin de continuer à utiliser la connexion sélectionnée, une fois que le composant a exécuté sa tâche.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate. **Trigger** : RunIf, OnComponentOk; OnComponentError, OnSubjobOk, OnSubjobError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate **Trigger** : Runif, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début et requiert un composant de sortie.	

Scénario 1 : Récupérer les métadonnées d'un système SAP

Les composants SAP (**tSAPInput** et **tSAPOutput**) ainsi que l'assistant SAP sont basés sur des bibliothèques validées et fournies par SAP (JCO) permettant à l'utilisateur d'appeler les fonctions RFC ou BAPI et de récupérer les données du système SAP au niveau des tables, RFC ou BAPI.

Ce scénario utilise l'assistant SAP guidant l'utilisateur à travers les différentes étapes de création de connexions SAP et des fonctions RFC et BAPI. L'assistant SAP est disponible uniquement pour les utilisateurs ayant souscrit à l'une des solutions *Talend*. Sinon, déposez de la **Palette** le composant **tSAPInput** et configurez ses propriétés manuellement.

Ce scénario utilise un assistant SAP pour créer une connexion au système SAP, appelle une fonction BAPI pour récupérer les informations d'une entreprise stockées dans le système SAP et les affiche dans le *Studio Talend*.

Les captures d'écran ci-dessous affichent les paramètres du système SAP permettant d'accéder aux informations sur l'entreprise que vous souhaitez lire dans le *Studio Talend* à l'aide du composant **tSAPInput**.

Paramétrer la connexion SAP à l'aide de l'assistant

Paramétrer la connexion au système SAP

1. Créez une connexion au système SAP à l'aide de l'assistant SAP. Dans ce scénario, la connexion SAP est appelée **sap** et a été sauvegardée dans le nœud **Metadata**.

2. Appelez la fonction BAPI appelée *BAPI_COMPANY_GETDETAIL* à l'aide de l'assistant SAP pour accéder au document HTML BAPI stocké dans le système SAP et consultez les informations sur l'entreprise.

3. Dans le champ **Name filter**, saisissez *BAPI** et cliquez sur le bouton **Search** pour afficher toutes les fonctions BAPI disponibles.

4. Sélectionnez *BAPI_COMPANY_GETDETAIL* pour afficher le schéma décrivant les informations de l'entreprise.

La vue à droite de l'assistant affiche les métadonnées de la fonction *BAPI_COMPANY_GETDETAIL* et vous permet de configurer les paramètres nécessaires.

La vue **Document** affiche le document HTML SAP de la fonction *BAPI_COMPANY_GETDETAIL*.

La vue **Parameter** fournit les informations sur les paramètres d'entrée et de sortie nécessaire à la fonction *BAPI_COMPANY_GETDETAIL* pour retourner les valeurs.

Définir les paramètres d'entrée et de sortie à l'aide de l'assistant

1. Dans la vue **Parameter**, cliquez sur l'onglet **Input** pour lister le ou les paramètre(s) d'entrée. Dans ce scénario, il n'y a qu'un seul paramètre d'entrée requis par la fonction *BAPI_COMPANY_GETDETAIL*. Ce paramètre s'appelle *COMPANYID*.

2. Dans la vue **Parameter**, cliquez sur l'onglet **Output** pour lister les paramètres de sortie retournés par la fonction *BAPI_COMPANY_GETDETAIL*. Dans ce scénario, il y a deux paramètres de sortie : *COMPANY_DETAIL* et *RETURN*.

Parameter type	Name	JCO type	Length	Value	Purpose:
structure	COMPANY_DETAIL		0		
single	COMPANY	CHAR	6		Company
single	NAME1	CHAR	30		Company name
single	NAME2	CHAR	30		Name of company 2
single	COUNTRY	CHAR	3		Country of company
single	LANGU	CHAR	1		Language key
single	STREET	CHAR	30		Street address of the company
single	PO_BOX	CHAR	10		Post office box of the company
single	POSTL_COD1	CHAR	10		Global company zip code
single	CITY	CHAR	30		City where company is located
single	CURRENCY	CHAR	5		Local currency
single	COUNTRY_ISO	CHAR	2		Country ISO code
single	CURRENCY_ISO	CHAR	3		ISO currency code
single	LANGU_ISO	CHAR	2		Language according to ISO 639
structure	RETURN		0		
single	TYPE	CHAR	1		Message type: S Success, E Error, '
single	CODE	CHAR	5		Message code
single	MESSAGE	CHAR	220		Message text
single	LOG_NO	CHAR	20		Application log: log number

Chacun des paramètres de "structure" est composé de nombreux paramètres "single".

La vue **Test it** permet d'ajouter ou de supprimer des paramètres d'entrée en fonction de la fonction appelée. Dans ce scénario, l'objectif est de récupérer les métadonnées du paramètre "structure" appelé *COMPANY_DETAIL* qui est composé de 14 paramètres "single".

3. Dans la colonne **Value** de la ligne **COMPANYID** du premier tableau, saisissez "*000001*" afin de retourner les données correspondant à l'entreprise dont l'identifiant est 000001.

4. Dans la liste **Output type** en bas de l'assistant, sélectionnez **output.table**.

5. Cliquez sur **Launch** en bas de la vue pour afficher la valeur de chaque paramètre "single" retourné par la fonction **BAPI_COMPANY_GETDETAIL**.

6. Cliquez sur **Finish** pour fermer l'assistant et créer la connexion.

La connexion **sap** et la fonction **BAPI_COMPANY_GETDETAIL** apparaissent sous le nœud **SAP Connections** du **Repository**.

Récupérer les différents schémas des fonctions SAP

Pour récupérer les différents schémas de la fonction **BAPI_COMPANY_GETDETAIL**, veuillez suivre la procédure suivante :

1. Cliquez-droit sur **BAPI_COMPANY_GETDETAIL** dans le **Repository** et sélectionnez **Retrieve schema** dans le menu.

2. Dans l'assistant qui s'ouvre, sélectionnez les schémas que vous souhaitez récupérer. Ici, sélectionnez les deux schémas : **COMPANY_DETAIL** et **RETURN**.

3. Cliquez sur **Next** pour afficher ces différents schémas puis cliquez sur **Finish** pour ferme la boîte de dialogue.

Les deux schémas apparaissent sous la fonction **BAPI_COMPANY_GETDETAIL** dans le **Repository.**

Récupérer les données correspondant à l'entreprise

Pour récupérer les données correspondant à l'entreprise dont l'identifiant est 000001 et les afficher dans le *Studio Talend* :

1. Dans le **Repository**, sélectionnez la connexion SAP que vous venez de créer et glissez-la dans l'espace de modélisation, sélectionnez **tSAPConnection** dans la liste de composants et cliquez sur **OK**. Le composant **tSAPConnection** portant le nom de votre connexion, *sap*, sera ajouté à votre Job et ouvrira une connexion SAP.

2. Double-cliquez sur le **tSAPConnection** pour afficher la vue **Component** et définir ses propriétés :

Si vous avez utilisé les métadonnées pour créer le composant, le mode **Repository** est sélectionné dans la liste **Property Type** et les champs suivants sont automatiquement renseignés. Si vous n'utilisez pas les métadonnées, sélectionnez le mode **Built-In** et renseignez les champs suivants manuellement.

3. Dans le **Repository**, déroulez le nœud **Metadata**, puis **sap** et glissez le schéma **RFC_READ_TABLE** dans l'espace de modélisation pour ouvrir une liste de composants.

4. Sélectionnez **tSAPInput** dans la liste de composants et cliquez sur **OK**.

5. Cliquez et déposez les composants **tFilterColumns** et **tLogRow** dans l'espace de modélisation.

6. Connectez le **tSAPConnection** au **tSAPInput** via un lien **Trigger** > **OnSubjobOk**

7. Pour connecter le **tSAPInput** au **tLogRow**, cliquez-droit sur le **tSAPInput**, sélectionnez **Row** puis **row_COMPANY_DETAIL_1** dans le menu et cliquez sur le composant **tLogRow**.

8. Dans l'espace de modélisation, sélectionnez le composant **tSAPInput** et cliquez sur la vue **Component** pour afficher ses propriétés.

   Les propriétés du composant **tSAPInput** apparaissent automatiquement puisque le schéma est stocké dans le dossier **Metadata** et que le composant est initialisé par l'assistant SAP.

9. Cochez la case **Use an existing connection** et sélectionnez le composant **tSAPConnection** dans la liste **Component list** pour utiliser la connexion ouverte par le composant de connexion, ici **sap**.

Dans la zone **Initialize input**, vous pouvez voir les paramètres d'entrée nécessaires à la fonction *BAPI_COMPANY_GETDETAIL*.

Dans la zone **Outputs**, vous pouvez voir les différents schémas de la fonction **BAPI_COMPANY_GETDETAIL**, notamment le schéma **COMPANY_DETAIL** que vous récupérez en sortie.

Exécuter le Job

1. Dans l'espace de modélisation, sélectionnez le composant **tLogRow** et cliquez sur la vue **Component** pour paramétrer ses propriétés. Pour plus d'informations, consultez *tLogRow*.

2. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

```
Starting job BAPI_SAP at 18:32 07/10/2009.

[statistics] connecting to socket on port 4100
[statistics] connected
.-----------------------------------.
|            #1. tLogRow_2          |
+---------------+-------------------+
| key           | value             |
+---------------+-------------------+
| COMPANY       | 000001            |
| NAME1         | Gesellschaft G00000 |
| NAME2         |                   |
| COUNTRY       | DE                |
| LANGU         | D                 |
| STREET        | Neurottstrasse 16 |
| PO_BOX        |                   |
| POSTL_COD1    | 69190             |
| CITY          | Walldorf          |
| CURRENCY      | EUR               |
| COUNTRY_ISO   | DE                |
| CURRENCY_ISO  | EUR               |
| LANGU_ISO     | DE                |
+---------------+-------------------+
[statistics] disconnected
Job BAPI_SAP ended at 18:32 07/10/2009. [exit code=0]
```

Le composant **tSAPInput** récupère du système SAP les métadonnées du paramètre *COMPANY_DETAIL* de type "structure" et le composant **tLogRow** affiche les informations dans la console.

Scénario 2 : Lire les données des différents schémas de la fonction RFC_READ_TABLE

Les composants SAP (**tSAPInput** et **tSAPOutput**) ainsi que l'assistant SAP sont basés sur des bibliothèques validées et fournies par SAP (JCO) permettant à l'utilisateur d'appeler les fonctions RFC ou BAPI et de récupérer les données du système SAP au niveau des tables, RFC ou BAPI.

Ce scénario utilise l'assistant SAP guidant l'utilisateur à travers les différentes étapes de création de connexions SAP et des fonctions RFC et BAPI. L'assistant SAP est disponible uniquement pour les utilisateurs ayant souscrit à l'une des solutions *Talend*. Sinon, déposez de la **Palette** le composant **tSAPInput** et configurez ses propriétés manuellement.

Ce scénario Java utilise l'assistant SAP pour créer une connexion au système SAP et pour appeler une fonction RFC permettant de lire directement dans une table appelée *SFLIGHT* à partir du système SAP. Puis, il affiche dans le *Studio Talend* les différentes structures de la table *SFLIGHT* stockée dans le système SAP.

Paramétrer et configurer la connexion SAP à l'aide de l'utilisateur

1. Créez une connexion au système SAP à l'aide de l'assistant SAP. Dans ce scénario, la connexion SAP est appelée **sap**.

2. Appelez la fonction RFC appelée *RFC_READ_TABLE* à l'aide de l'assistant SAP pour accéder à la table du système SAP et pour visualiser sa structure.

3. Dans le champ **Name filter**, saisissez *RFC** et cliquez sur le bouton **Search** pour afficher toutes les fonctions RFC disponibles.

4. Sélectionnez *RFC_READ_TABLE* pour afficher les schémas de cette fonction.

La vue à droite de l'assistant affiche les métadonnées de la fonction *RFC_READ_TABLE* et permet de définir les paramètres nécessaires.

L'onglet **Document** affiche le document html SAP de la fonction *RFC_READ_TABLE*.

L'onglet **Parameter** fournit les informations sur les paramètres requis par la fonction *RFC_READ_TABLE* pour pouvoir retourner les valeurs des paramètres.

Définir les paramètres d'entrée et de sortie à l'aide de l'assistant

1. Dans la vue **Parameter**, cliquez sur l'onglet **Table** pour obtenir une description de la structure des différentes tables de la fonction *RFC_READ_TABLE*.

Parameter type	Name	JCO type	Length	Value	Purpose:
table	DATA		0		
single	WA	CHAR	512		Character field length 512
table	FIELDS		0		
single	FIELDNAME	CHAR	30		Field name
single	OFFSET	NUM	6		Offset of a field in work area
single	LENGTH	NUM	6		Length (no. of characters)
single	TYPE	CHAR	1		ABAP data type (C,D,N,...)
single	FIELDTEXT	CHAR	60		Short text describing R/3 Rep
table	OPTIONS		0		
single	TEXT	CHAR	72		Text line of a message

La vue **Test it** vous permet d'ajouter ou de supprimer des paramètres d'entrée en fonction de la fonction appelée. Dans ce scénario, l'objectif est de récupérer les différentes structures de la table *SFLIGHT*.

Name	Parameter Type	JCO type	Structure|Table	Length	Value	Purpose:
DELIMITER	input.single	CHAR		1	;	Sign for indicating field limits in DATA
NO_DATA	input.single	CHAR		1		If <> SPACE, only FIELDS is filled
QUERY_TABLE	input.single	CHAR		30	SFLIGHT	Table read
ROWCOUNT	input.single	INT		4		If <> SPACE, only FIELDS is filled
ROWSKIPS	input.single	INT		4		If <> SPACE, only FIELDS is filled

Name	Parameter Type	JCO type	Structure|Table	Length	Value	Purpose:
WA	table.output	CHAR	DATA	512	000;LH ;0400;199...	Character field length 512
FIELDNAME	table.output	CHAR	FIELDS	30	MANDT	Field name
OFFSET	table.output	NUM	FIELDS	6	000000	Offset of a field in work...
LENGTH	table.output	NUM	FIELDS	6	000003	Length (no. of charact...
TYPE	table.output	CHAR	FIELDS	1	C	ABAP data type (C,D,N...
FIELDTEXT	table.output	CHAR	FIELDS	60	Client for WB train...	Short text describing R...
TEXT	table.output	CHAR	OPTIONS	72		Text line of a message

2. Dans la colonne **Value** de la ligne **DELIMITER** du premier tableau, saisissez ";" comme séparateur de champ.

3. Dans la colonne **Value** de la ligne **QUERY_TABLE**, saisissez *SFLIGHT* comme table à requêter.

4. Dans la liste **Output type** en bas de l'assistant, sélectionnez **output.table**.

5. Dans la liste **Constructure|Table**, sélectionnez **DATA**.

6. Cliquez sur **Launch** en bas de la vue pour afficher les valeurs de paramètres retournées par la fonction *RFC_READ_TABLE*. Dans ce scénario, le délimiteur est ";" et la table à lire est *SFLIGHT*.

7. Cliquez sur **Finish** pour fermer l'assistant et créer la connexion.

Récupérer les différents schémas de la fonction RFC_READ_TABLE

La connexion **sap** et la fonction **RFC_READ_TABLE** apparaissent sous le nœud **SAP Connections** du **Repository**.

Pour récupérer les différents schémas de la fonction **RFC_READ_TABLE** :

1. Cliquez-droit sur **RFC_READ_TABLE** dans le **Repository**, et sélectionnez **Retrieve schema** dans le menu.

2. Dans l'assistant qui s'ouvre, sélectionnez les schémas que vous souhaitez récupérer. Ici, sélectionnez les trois schémas : **DATA**, **FIELDS** et **OPTIONS**.

3. Cliquez sur **Next** pour afficher ces différents schémas.

4. Cliquez sur **Finish**.

Les trois schémas apparaissent sous la fonction **RFC_READ_TABLE** dans le **Repository.**

Récupérer les noms des colonnes de données du schéma SFLIGHT

Dans ce scénario, l'objectif est de récupérer les données et le nom des colonnes de la table **SFLIGHT** pour les récupérer et les afficher dans le *Studio Talend*. Pour ce faire, suivez la procédure suivante :

Construire le Job

1. Dans le **Repository**, sélectionnez la fonction **RFC_READ_TABLE** de la connexion *sap* et glissez-la dans l'espace de modélisation, sélectionnez **tSAPInput** dans la liste de composants et cliquez sur **OK**. Le **tSAPInput** s'affiche dans l'espace de modélisation.

2. Cliquez et déposez deux composants **tLogRow** dans l'espace de modélisation.

3. Cliquez-droit sur le **tSAPInput**, sélectionnez **Row** puis **row_DATA_1** dans le menu et cliquez sur le premier composant **tLogRow**.

4. Cliquez-droit sur le **tSAPInput**, sélectionnez **Row** puis **row_FIELDS_1** dans le menu et cliquez sur le deuxième composant **tLogRow**.

L'objectif de ce scénario est de récupérer les schémas **FIELDS** et **DATA** et les mettre dans deux flux de sortie différents.

5. Dans l'espace de modélisation, double-cliquez sur **tSAPInput** pour afficher ses propriétés dans la vue **Component**.

Les propriétés du composant **tSAPInput** apparaissent automatiquement puisque le schéma est stocké dans le dossier **Metadata** et que le composant est initialisé par l'assistant SAP.

Dans la zone **Initialize input**, vous pouvez voir les paramètres d'entrée nécessaires pour la fonction *RFC_READ_TABLE*, à savoir le séparateur de champ ";" et le nom de la table **SFLIGHT**.

Dans la zone **Outputs**, vous pouvez voir les différents schémas de la table **SFLIGHT**.

Outputs

Schema	Type(for iterate)	TableName...	Mapping
row_FIELDS_1	table_output	"FIELDS"	"FIELDNAME","OFFSET
row_OPTIONS_1	table_output	"OPTIONS"	"TEXT"
row_DATA_1	table_output	"DATA"	"WA"

Exécuter le job

1. Dans l'espace de modélisation, sélectionnez les composants **tLogRow** et cliquez sur la vue **Component** pour paramétrer leurs propriétés. Pour plus d'informations, consultez *tLogRow*.

2. Appuyez sur **CTRL+S** pour enregistrer votre Job, puis sur **F6** pour l'exécuter.

```
Starting job jobsap at 16:13 01/10/2009.

[statistics] connecting to socket on port 4093
[statistics] connected
.----------+------+------+----+----------------------------------------.
|                              tLogRow_3                                |
|=---------+------+------+----+----------------------------------------=|
|FIELDNAME |OFFSET|LENGTH|TYPE|FIELDTEXT                               |
|=---------+------+------+----+----------------------------------------=|
|MANDT     |0     |3     |C   |Client for WB train. data model BC_Travel|
|CARRID    |4     |3     |C   |Airline carrier ID                      |
|CONNID    |8     |4     |N   |Flight connection Id                    |
|FLDATE    |13    |8     |D   |Flight date                             |
|PRICE     |22    |15    |P   |Airfare                                 |
|CURRENCY  |38    |5     |C   |Local currency of airline               |
|PLANETYPE |44    |10    |C   |Plane type                              |
|SEATSMAX  |55    |10    |X   |Maximum capacity                        |
|SEATSOCC  |66    |10    |X   |Occupied seats                          |
|PAYMENTSUM|77    |17    |P   |Total of current bookings               |
'----------+------+------+----+----------------------------------------'

.-------------------------------------------------------------------------.
|                                  tLogRow_2                              |
|=------------------------------------------------------------------------=|
|WA                                                                       |
|=------------------------------------------------------------------------=|
|000;LH ;0400;19950228; 899.00 ;DEM  ;A319      ;5E010000  ;03000000  ; 2639.00|
|000;LH ;0454;19951117;1499.00 ;DEM  ;A319      ;5E010000  ;02000000  ; 2949.00|
|000;LH ;0455;19950606;1090.00 ;USD  ;A319      ;DC000000  ;01000000  ; 1499.00|
|000;LH ;3577;19950428;6000.00 ;LIT  ;A319      ;DC000000  ;01000000  ;  600.00|
|000;SQ ;0026;19950228; 849.00 ;DEM  ;DC-10-10  ;7C010000  ;02000000  ; 1684.00|
'-------------------------------------------------------------------------'

[statistics] disconnected
Job jobsap ended at 16:13 01/10/2009. [exit code=0]
```

Le composant **tSAPInput** récupère le nom des colonnes de la table *SFLIGHT* et les données correspondantes à partir du système SAP et les composants **tLogRow** les affichent sous forme de table dans la console.

Scénario 3 : Récupérer les données d'un système SAP

Ce scénario décrit un Job qui ouvre une connexion à un système SAP avec tous les paramètres de connexion configurés dans le composant **tSAPConnection**, récupère les données stockées dans une table du système SAP et affiche les données dans la console.

Déposer les composants

- Créez un nouveau Job et ajoutez les trois composants indiqués ci-dessous en saisissant leur nom dans l'espace de modélisation ou en les déposant de la **Palette** :
  - le **tSAPConnection** : ouvre une connexion au système SAP.
  - le **tSAPInput** : récupère les données dans une table du système SAP.
  - le **tLogRow** : affiche les données qu'il reçoit du composant précédent dans la console.

Configurer le composant de connexion

1. Double-cliquez sur le composant **tSAPConnection** afin d'ouvrir sa vue **Basic settings**.

2. Sélectionnez la version de SAP Java Connector que vous utilisez dans la liste déroulante **SAP JCO Version**. Dans cet exemple, il s'agit de *sap jco 3.**.

3. Renseignez les champs de configuration de connexion avec vos informations de connexion au système SAP. Pour plus de détails concernant les paramètres de connexion, consultez le *tSAPConnection*

Configurer le composant d'entrée

1. Double-cliquez sur le composant **tSAPInput** afin d'afficher sa vue **Basic settings**.

2. Cochez la case **Use an existing connection**, puis sélectionnez la connexion que vous avez configurée auparavant dans la liste déroulante **Connection configuration**. Dans cet exemple, il s'agit de *tSAPConnection_1*.

   Si vous n'utilisez pas de composant **tSAPConnection**, vous devez renseigner tous les paramètres de connexion dans les champs du composant **tSAPInput** prévus à cet effet.

3. Dans le champ **FunName**, saisissez le nom de la fonction SAP. Dans cet exemple, il s'agit de la fonction *RFC_READ_TABLE* qui permet de lire les données de la table SAP.

4. Dans la table **Initialize input**, cliquez deux fois sur le bouton **[+]** pour ajouter deux paramètres d'entrée (indiqués ci-dessous) requis par la fonction à laquelle vous faites appel, puis définissez leurs valeurs **ParameterValue**, **Type** et **ParameterName**.

   - *DELIMITER* : le séparateur de champ dans les données écrites. Il s'agit d'un paramètre de type *input_single*, *";"* dans cet exemple.

   - *QUERY_TABLE* : le nom de la table interrogée. Il s'agit d'un paramètre de type *input_single*,*"SFLIGHT"* dans cet exemple.

5. Dans la table **Outputs**, cliquez sur le bouton **[+]** pour ajouter une ligne au schéma des données écrites retournées par la fonction appelée.

6. Dans la cellule **TableName** de la nouvelle ligne ajoutée, saisissez le nom de la table à partir de laquelle récupérer les données. Dans cet exemple, il s'agit de la table *"DATA"*.

7. Dans la cellule **Type** de la nouvelle ligne ajoutée, sélectionnez *table_output* dans la liste déroulante.

8. Sélectionnez la cellule **Schema** de la nouvelle ligne ajoutée et cliquez sur le bouton **[...]** dans la cellule pour ouvrir la boîte de dialogue **[Give the name for the schema]**. Renseignez le champ **Schema Name**,*output* dans cet exemple.

Cliquez sur **OK** pour fermer la boîte de dialogue.

9. Dans la boîte de dialogue qui s'ouvre, cliquez sur le bouton **[+]** pour ajouter une colonne, *WA* de type String dans cet exemple.

Cliquez sur **OK** pour sauvegarder les paramètres.

10. Sélectionnez la cellule **Mapping** de la nouvelle ligne ajoutée et cliquez sur le bouton **[...]** dans la cellule pour ouvrir la boîte de dialogue **Mappings**. Dans la cellule **Schema XPathQuerys**, saisissez la requête XPath. Dans cet exemple, il s'agit de *"WA"*.

Configurer le composant de sortie

1. Double-cliquez sur le composant **tLogRow** afin d'ouvrir sa vue **Basic settings**.

2. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)**.

Relier les composants et exécuter le Job

1. Liez le composant **tSAPConnection** au composant **tSAPInput** à l'aide d'une connexion **Trigger** > **On Subjob Ok**, puis reliez le composant **tSAPInput** au composant **tLogRow** à l'aide d'une connexion **Row** > **output**.

2. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.

3. Appuyez sur **F6** pour exécuter le Job.

```
[statistics] connecting to socket on port 3862
[statistics] connected
.-------------------------------------------------------------------------------------------.
|                                          tLogRow_1                                        |
|=-----------------------------------------------------------------------------------------=|
|WA                                                                                         |
|=-----------------------------------------------------------------------------------------=|
|000;LH ;0400;19950228;        899.00 ;DEM  ;A319      ;5E010000  ;03000000  ;        2639.00|
|000;LH ;0454;19951117;       1499.00 ;DEM  ;A319      ;5E010000  ;02000000  ;        2949.00|
|000;LH ;0455;19950606;       1090.00 ;USD  ;A319      ;DC000000  ;01000000  ;        1499.00|
|000;LH ;3577;19950428;       6000.00 ;LIT  ;A319      ;DC000000  ;01000000  ;         600.00|
|000;SQ ;0026;19950228;        849.00 ;DEM  ;DC-10-10  ;7C010000  ;02000000  ;        1684.00|
'-------------------------------------------------------------------------------------------'

[statistics] disconnected
```

Les données correspondantes dans l'un des schémas *DATA* de la table *SFLIGHT* sont affichées dans la console.

tSAPOutput

Propriétés du tSAPOutput

Famille de composant	Business	Ce composant est déprécié.
Fonction	Le composant **tSAPOutput** écrit dans un système SAP.	
Objectif	Il permet d'écrire des données dans un système SAP.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant connexion de base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	SAP JCO Version	Sélectionnez la version de SAP Java Connector dans la liste déroulante.
	Connection configuration	**Client type :** Saisissez votre code de connexion SAP habituel. **Userid :** Saisissez l'identifiant de connexion utilisateur. **Password :** Saisissez le mot de passe. **Language :** Spécifiez la langue. **Host name :** Saisissez l'adresse IP du système SAP. **System number :** Saisissez le numéro du système. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	FunName	Saisissez le nom de la fonction que vous souhaitez utiliser pour récupérer les données.
	Mapping	Configurez les paramètres afin de sélectionner les données à écrire dans le système SAP.
	Specify a shared destination	Cochez cette case et spécifiez la destination partagée définie du côté *Talend Runtime* dans le champ **Shared destination name**. Le champ **Shared destination name** permet aux composants tSAP* d'utiliser une destination partagée dans *Talend Runtime*, lorsqu'ils sont exécutés comme partie d'une Route ESB ou d'un service de données ESB dans l'environnement OSGi du conteneur de *Talend Runtime*. Cette option est disponible uniquement lorsque vous sélectionnez **sap jco 3.*** dans la liste déroulante **SAP JCO Version** et que vous ne cochez pas la case **Use an existing connection**. En plus de spécifier la destination partagée, dans le champ **Shared destination name** du composant, vous devez également configurer la source des données SAP (DataSource) pour *Talend Runtime*. La procédure pour définir la source des données SAP et pour la configurer sont décrites dans le fichier *lisezmoi.txt* fourni dans le package de Talend ESB, dans le dossier */add-ons/datasource/sap*. *Si vous utilisez la configuration propre du composant, votre connexion SAP se ferme à la fin de l'exécution du composant. Pour éviter cela, utilisez une destination partagée ayant un nom spécifié.*
Advanced settings	*SAP Properties*	Si vous souhaitez utiliser une configuration personnalisée pour le système SAP à utiliser, renseignez cette table avec la ou les propriété(s) à personnaliser. Par exemple, • si vous devez établir une connexion Unicode entre le système SAP et le *Studio Talend*, ajoutez la propriété *jco.server.unicode* et configurez sa valeur à *1*. • si vous devez configurer une connexion SAP avec SNC (Secure Network Communications), ajoutez les paramètres SNC et configurez leur valeur dans cette table. Pour plus d'informations concernant la configuration des paramètres SCN, consultez http://help.sap.com/saphelp_erp60_sp/helpdata/en/ce/1dfd3d4aefd95ee10000000a114084/content.htm et https://help.sap.com/javadocs/NW04/current/jc/com/sap/mw/jco/JCO.html (en anglais).
	ftp and http based programs	Pour invoquer du serveur SAP une fonction nécessitant le téléchargement d'un document, cochez cette case et assurez-vous que SAPGUI a bien été installé avec votre système SAP. Si la case est cochée mais que SAPGUI n'est pas installé, des erreurs vont subvenir. *Cette case est indisponible si vous cochez la case **Use an existing connection** dans l'onglet **Basic settings**.*

	Release Connection	Décochez cette case afin de continuer à utiliser la connexion sélectionnée, une fois que le composant a exécuté sa tâche.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de sortie et requiert un composant d'entrée.	

Scénarios associés

Aucun scénario n'est disponible pour ce composant.

tSAPRollback

Propriétés du tSAPRollback

Famille de composant	Business/SAP	
Fonction	Le composant **tSAPRollback** annule la transaction dans le SAP connecté.	
Objectif	Le **tSAPRollback** évite le commit de transaction involontaire.	
Basic settings	*SAPConnection Component list*	Sélectionnez le composant de connexion **tSAPConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Release Connection	Décochez cette case afin de continuer à utiliser la connexion sélectionnée, une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion SAP parmi celles prévues dans votre Job. Lorsqu'un paramètre dynamique est défini, l'option **SAPConnection Component List** correspondante est désactivée dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants SAP et notamment avec les composants **tSAPConnection** et **tSAPCommit**.	

Scénario associé

Pour un scénario associé au **tSAPRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMysqlRollback**.

tSugarCRMInput

Propriétés du tSugarCRMInput

Famille du composant	Business/Cloud	
Fonction	Le **tSugarCRMInput** se connecte à un module de la base de données SugarCRM via le service Web adéquat.	
Objectif	Le **tSugarCRMInput** permet d'extraire les données d'une base de données SugarCRM à l'aide d'une requête.	
Basic settings	*SugarCRM Webservice URL*	Saisissez l'URL du service Web permettant de se connecter à la base de données SugarCRM.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Module	Sélectionnez le module adéquat dans la liste. Pour utiliser des tables personnalisées, sélectionnez **Use custom module** dans la liste . Les champs **Custom module package name** et **Custom module name** qui s'affichent sont automatiquement remplis avec les noms correspondants.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Pour ce composant, le schéma dépend du **Module** sélectionné.
	Query condition	Saisissez la requête permettant de sélectionner les données à extraire. Exemple : account_name= ‘Talend'.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire.

Scénario: Extraire les données d'un compte à partir de SugarCRM

Ce scénario décrit un Job à deux composants visant à extraire les informations d'un compte à partir d'une base de données SugarCRM vers un fichier de sortie Excel.

Configurer le Job

1. A partir de la **Palette**, cliquez-déposez un composant **tSugarCRMInput** et un composant **tFileOutputExcel**.
2. Connectez les composants d'entrée et de sortie à l'aide d'un lien de type **Row** > **Main**.

Configurer le composant d' entrée

1. Double-cliquez sur le composant **tSugarCRMInput** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.

2. Remplissez les informations de connexion dans les champs **SugarCRM Web Service URL**, **Username** et **Password**.

3. Sélectionnez ensuite le module adéquat parmi ceux proposés dans la liste **Module**. Dans cet exemple, c'est *Accounts* qui est sélectionné.

   Le champ **Schema** est automatiquement renseigné en fonction du module sélectionné, mais il est possible de le modifier et de supprimer les colonnes dont vous n'avez pas besoin en sortie.

4. Dans le champ **Query Condition**, saisissez la requête que vous voulez utiliser pour extraire les données du CRM. Dans cet exemple : "*billing_address_city='Sunnyvale'*".

Exécuter le Job

1. Sélectionnez ensuite le composant **tFileOutputExcel**.

2. Renseignez le nom du fichier de destination dans le champ **File Name** ainsi que le nom de la feuille de calcul dans le champ **Sheet** puis cochez la case **Include header** pour considérer les entêtes.

3. Enregistrez votre Job puis appuyez sur **F6** pour l'exécuter.

   Les données filtrées sont inscrites en sortie dans la feuille de calcul du fichier Excel spécifié.

tSugarCRMOutput

Propriétés du tSugarCRMOutput

Famille du composant	Business/Cloud	
Fonction	Le composant **tSugarCRMOutput** écrit dans un module de la base de données SugarCRM via le service Web adéquat.	
Objectif	Le **tSugarCRMOutput** permet d'écrire des données dans une base de données SugarCRM.	
Basic settings	*SugarCRM WebService URL*	Saisissez l'URL du service Web permettant de se connecter à la base de données SugarCRM.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Module	Sélectionnez le module adéquat dans la liste. Pour utiliser des tables personnalisées, sélectionnez **Use custom module** dans la liste . Les champs **Custom module package name** et **Custom module name** qui s'affichent sont automatiquement remplis avec les noms correspondants.
	Action	Vous pouvez effectuer l'une des opérations suivantes sur les données du module SugarCRM : **Insert** : insérer les données. **Update** : mettre les données à jour.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tVtigerCRMInput

Propriétés du tVtigerCRMInput

Famille du composant	Business/VtigerCRM	
Fonction	Le **tVtigerCRMInput** se connecte à un module de la base de données vTigerCRM via le service Web adéquat.	
Objectif	Le **tVtigerCRMInput** permet d'extraire les données d'une base de données tVtigerCRM.	
Basic settings		
Vtiger Version	Sélectionnez la version du Vtiger Web Services que vous souhaitez utiliser.	
Vtiger 5.0	*Server Address*	Saisissez l'adresse IP du serveur VtigerCRM.
	Port	Saisissez le numéro de port d'accès au serveur.
	Vtiger Path	Saisissez le chemin d'accès au serveur
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Version	Renseignez la version de VtigerCRM que vous utilisez.
	Module	Sélectionnez le module adéquat dans la liste.
	Method	Sélectionnez la méthode adéquate dans la liste. Cette méthode permet de déterminer l'action à appliquer au module VtigerCRM sélectionné.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Pour ce composant, le schéma dépend du **Module** sélectionné.
Vtiger 5.1	*Endpoint*	Saisissez l'URL du serveur Web invoqué.
	Username	Saisissez le nom d'utilisateur afin de vous connecter au VtigerCRM..
	Access key	Saisissez la clé d'accès pour le nom d'utilisateur.

	Query condition	Saisissez la requête permettant de sélectionner les données à extraire.
	Manual input of SQL query	Cochez cette case pour saisir manuellement la requête que vous souhaitez effectuer dans le champ **Query**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tVtigerCRMOutput

Propriétés du tVtigerCRMOutput

Famille du composant	Business/VtigerCRM	
Fonction	Le composant **tVtigerCRMOutput** écrit dans un module de la base de données VtigerCRM via le service Web adéquat.	
Objectif	Le **tVtigerCRMOutput** permet d'écrire des données dans une base de données VtigerCRM.	
Basic settings		
Vtiger Version	Sélectionnez la version du Vtiger Web Services que vous souhaitez utiliser.	
Vtiger 5.0	*Server Address*	Saisissez l'adresse IP du serveur VtigerCRM.
	Port	Saisissez le numéro de port d'accès au serveur.
	Vtiger Path	Saisissez le chemin d'accès au serveur.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur au service Web. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Version	Renseignez la version de VtigerCRM que vous utilisez.
	Module	Sélectionnez le module adéquat dans la liste.
	Method	Sélectionnez la méthode adéquate dans la liste. Cette méthode permet de déterminer l'action à appliquer au module VtigerCRM sélectionné.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Pour ce composant, le schéma dépend du **Module** sélectionné.
Vtiger 5.1	*Endpoint*	Saisissez l'URL du serveur Web invoqué.
	Username	Saisissez le nom d'utilisateur afin de vous connecter au VtigerCRM.
	Access key	Saisissez la clé d'accès pour le nom d'utilisateur.

	Action	Vous pouvez effectuer l'une des opérations suivantes sur les données du module VtigerCRM : **Insert** : insérer les données. **Update** : mettre les données à jour.
	Module	Sélectionnez le module adéquat dans la liste.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Pour ce composant, le schéma dépend du **Module** sélectionné.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

Composants Business Intelligence

Ce chapitre passe en revue les principaux composants présents dans la famille **Business Intelligence** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **BI** regroupe tous les connecteurs couvrant des besoins tels que la lecture et l'écriture de bases de données multidimensionnelles ou OLAP, la génération de rapports Jasper, le suivi des modifications de bases de données à l'aide de tables Slow Changing Dimension, etc.

tBarChart

Propriétés du tBarChart

Famille du composant	Business Intelligence/Charts	
Fonction	Le composant **tBarChart** lit des données à partir d'un flux d'entrée et transforme les données en diagramme en barres dans un fichier image au format PNG.	
Objectif	Le composant **tBarChart** génère un diagramme en barres à partir des données d'entrée afin de faciliter l'analyse technique.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Le schéma du **tBarChart** contient trois colonnes en lecture seule nommées respectivement **series** (type String), **category** (type String), et **value** (type Integer), selon un ordre prédéterminé. Les données de toutes les autres colonnes seront simplement passées au composant suivant, s'il y en a un, mais elles ne seront pas présentes dans le diagramme en barres.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Permet de synchroniser le schéma du fichier d'entrée avec le schéma du fichier de sortie. La fonction Sync ne s'affiche que lorsqu'une connexion de type **Row** est connectée au composant de sortie.
	Generated image path	Nom et chemin d'accès du fichier image de sortie.
	Chart title	Saisissez le titre du diagramme en barres à générer.
	Include legend	Cochez cette case si vous souhaitez inclure une légende à votre diagramme en barres, afin d'afficher toutes les séries (*series*) dans des couleurs différentes.
	3Dimensions	Cochez cette case afin de créer une image en trois dimensions. Par défaut, cette case est cochée et les barres représentant les séries (*series*) de chaque catégorie (*category*) sont empilées les unes sur les autres. Si cette case est décochée, une image en deux dimensions est

		créée, et les barres s'affichent les unes à côté des autres le long de l'axe *category*.
	Image width et *Image height*	Saisissez la largeur et la hauteur du fichier image, exprimées en pixels.
	Category axis name et *Value axis name*	Saisissez le nom de l'axe *category* et le nom de l'axe *value*.
	Foreground alpha	Saisissez un entier entre *0* et *100* afin de définir la transparence de l'image. Plus ce nombre est petit, plus votre image sera transparente.
	Plot orientation	Choisissez l'orientation qu'aura votre diagramme en barres: **VERTICAL** ou **HORIZONTAL**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant d'entrée ainsi qu'une connexion de type **Row Main**.	

Scénario : Créer un diagramme en barres à partir de données d'entrée

Ce scénario décrit un Job simple qui lit les données d'un fichier CSV et les transforme en diagramme en barres. Le fichier d'entrée est présenté ci-dessous :

```
City;Population(x1000);LandArea(km2);PopulationDensity(people/km2)
Beijing;10233;1418;7620
Moscow;10452;1081;9644
Seoul;10422;605;17215
Tokyo;8731;617;14151
Jakarta;8490;664;12738
New York;8310;789;10452
```

Le fichier d'entrée ayant une structure différente de celle requise pour l'utilisation d'un **tBarChart**, un **tMap** sera utilisé dans ce scénario, afin de mettre en correspondance les données avec un fichier CSV à trois colonnes, avant d'utiliser le **tBarChart** pour générer un fichier représentant le diagramme en barres.

Le **tMap** est généralement utilisé afin d'adapter le schéma d'entrée conformément à la structure du schéma du composant **tBarChart**. Pour plus d'informations concernant le composant **tMap**, consultez le *Guide utilisateur du Studio Talend* ainsi que *tMap*.

Déposer et relier les composants

1. A partir de la **Palette**, déposez les composants suivants dans l'espace de modélisation graphique : deux composants **tFileInputDelimited**, un **tMap**, trois composants **tFileOutputDelimited** et un **tBarChart**. Renommez les composants comme vous le souhaitez pour faciliter le travail de conception.

2. Connectez le composant **tPrejob** à un **tFileDelete** à l'aide d'une connexion **Trigger** > **On Component Ok** et connectez le composant **tPostjob** au second **tFileDelete** via le même type de connexion.

3. Connectez le premier composant **tFileInputDelimited** au **tMap** à l'aide d'une connexion **Row** > **Main**.

4. Connectez le composant **tMap** au premier **tFileOutputDelimited** à l'aide d'une connexion **Row** > **Main** et appelez-la *Population*.

5. Répétez les étapes ci-dessus pour connecter le composant **tMap** aux deux autres **tFileOutputDelimited** à l'aide d'une connexion **Row** > **Main** et appelez-les *Area* et *Density*.

6. Connectez le **tFileInputDelimited** au composant **tBarChart** à l'aide d'un lien **Row** > **Main**.

7. Connectez le premier composant **tFileInputDelimited** au second **tFileInputDelimited** en utilisant une connexion **Trigger** > **OnSubjobOk**.

8. Renommez les composants de façon à décrire leur fonction.

Lire les données source

1. Double-cliquez sur le premier composant **tFileInputDelimited** afin d'afficher l'onglet **Basic settings** de sa vue **Component**.

2. Renseignez le chemin d'accès au fichier à traiter dans le champ **File name**.

3. Spécifiez dans le champ **Header** le nombre de lignes d'en-tête. Dans ce scénario, il n'y a qu'une seule ligne.

4. Cliquez sur **Edit schema** pour décrire la structure des données du fichier d'entrée. Dans ce scénario, le schéma d'entrée est composé de quatre colonnes : *City*, *Population*, *Area*, et *Density*. Une fois les noms de colonnes et les types de données définis, cliquez sur **OK** pour fermer la boîte de dialogue du schéma.

Adapter les données source au schéma tBarChart

1. Double-cliquez sur le **tMap** pour ouvrir l'éditeur **Map Editor**.

   Vous pouvez voir une table d'entrée dans le panneau d'entrée, *row1* dans cet exemple et dans le panneau de sortie, trois tables de sortie vides appelées *Population*, *Area* et *Density*.

2. Utilisez l'éditeur **Schema editor** pour ajouter trois colonnes à chacune des tables de sortie : *series* (type String), *category* (type String) et *value* (type Integer).

3. Dans le champ **Expression** de chaque table de sortie, saisissez le texte à afficher dans la légende du graphique, dans cet exemple `"Population (x1000 people)"`, `"Land area (km2)"`, et `"Population density (people/km2)"`.

4. Déposez la colonne *City* de la table d'entrée dans la colonne *category* de chaque table de sortie.

5. Déposez la colonne *Population* de la table d'entrée dans la colonne *value* de la table *Population*.

6. Déposez la colonne *Area* de la table d'entrée dans la colonne *value* de la table *Area*.

7. Déposez la colonne *Density* de la table d'entrée dans la colonne *value* de la table *Density*.

8. Cliquez sur **OK** pour sauvegarder les mappings, fermer l'éditeur et propager les schémas de sortie aux composants de sortie.

Générer le fichier d'entrée temporaire

1. Double-cliquez sur le premier composant **tFileOutputDelimited** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **File Name**, définissez un fichier CSV dans lequel envoyer les flux de données mappés. Dans cet exemple, nommez le fichier de sortie à créer *Temp.csv*. Ce fichier sera utilisé comme fichier d'entrée pour le composant **tBarChart**.

3. Cochez la case **Append**.

4. Définissez les paramètres des deux autres composants **tFileOutputDelimited** exactement de la même manière que le premier **tFileOutputDelimited.**

Notez que l'ordre des flux de données sortant du composant **tMap** n'est pas nécessairement l'ordre dans lequel les données seront écrites dans le composant cible. Pour vous assurer que le fichier est correctement généré, supprimez,

s'il existe, le fichier portant le même nom avant d'exécuter le Job. Vérifiez que la case **Append** est cochée dans tous les composants **tFileOutputDelimited** afin que tous les flux de données mappés aillent dans le même fichier, sans écraser les données existantes.

Configurer les données d'entrée pour le tBarChart

1. Double-cliquez sur le second composant **tFileInputDelimited** afin d'afficher l'onglet **Basic settings** de sa vue **Component**.

2. Renseignez le champ **File name** avec le chemin d'accès au fichier, ainsi que son nom, comme défini dans chaque composant **tFileOutputDelimited**. Dans cet exemple, le fichier d'entrée est *LargeCities_mapped.csv*.

3. Double-cliquez sur le composant **tBarChart** pour afficher l'onglet **Basic settings**.

4. Dans le champ **Generated image path**, renseignez le chemin d'accès du fichier image à générer.

5. Dans le champ **Chart title**, saisissez un titre pour le diagramme en barres.

6. Saisissez un nom pour les axes *category* et *series*.

7. Si nécessaire, définissez la taille (la largeur dans le champ **Image Width**, la hauteur dans le champ **Image height**) et le degré de transparence de l'image (dans le champ **Foreground alpha**). Dans ce scénario, laissez les paramètres par défaut.

8. Cliquez sur **Edit schema** afin d'ouvrir la boîte de dialogue du schéma.

9. Copiez toutes les colonnes du schéma de sortie dans le schéma d'entrée en cliquant sur la double-flèche pointant vers la gauche.

   Cliquez sur **OK** pour fermer la boîte de dialogue du schéma.

Supprimer le fichier temporaire

La fonction des composants **tPrejob** et **tPostjob** est de déclencher les sous-jobs connectés et n'ont pas besoin d'être configurés. Vous devez simplement définir les propriétés des deux composants **tFileDelete**.

1. Double-cliquez sur le premier composant **tFileDelete** pour afficher sa vue **Basic settings**.

2. Dans le champ **File name**, renseignez le chemin d'accès du fichier temporaire.

   Si la case **Fail on error** est cochée et que le sous-job de pré-traitement échoue à cause d'erreurs (par exemple le fichier à supprimer n'existe pas), l'échec empêche le lancement du sous-job principal. Dans cette situation, décochez la case **Fail on error** pour éviter cette interruption.

Exécuter votre Job

1. Sauvegardez votre Job.
2. Appuyez sur **F6** pour l'exécuter.

   Un diagramme en barres est généré selon les critères définis.

tDB2SCD

Propriétés du tDB2SCD

Famille du composant	Databases/Business Intelligence/DB2	
Fonction	Le composant **tDB2SCD** reflète et traque les modifications d'une table DB2 SCD dédiée.	
Objectif	Le **tDB2SCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Property type	Le schéma peut être **Built-in** ou distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Table schema	Nom du schéma de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving Mode	Cochez cette case pour améliorer les performances du système.
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Additional JDBC parameters*	Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.
	End date time details	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.	

Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tDB2SCDELT

Propriétés du tDB2SCDELT

Famille du composant	Databases/Business Intelligence/DB2	
Fonction	Le composant **tDB2SCDELT** reflète et traque les modifications d'une table DB2 SCD dédiée.	
Objectif	Le **tDB2SCDELT** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table DB2 SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Source table	Nom de la table contenant les données à filtrer.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois pour que l'opération d'insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.

		Clear a table : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Surrogate Key	Sélectionnez dans la liste une colonne à utiliser comme clé de substitution.
	Creation	Sélectionnez la méthode à utiliser pour générer la clé de substitution. Pour plus d'informations concernant les méthodes de création, consultez *Clés SCD*.
	Source Keys	Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l'unicité des données entrantes.
	Source fields value include Null	Cochez cette case pour autoriser les valeurs Nulls dans les colonnes source. Les colonnes source font référence aux champs définis dans les tables **SCD type 1 fields** et **SCD type 2 fields**.
	Use SCD Type 1 fields	Utilisez le type 1 si vous n'avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.
	Use SCD Type 2 fields	Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications. **Start date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l'une des colonnes d'entrée du schéma comme date de départ (**Start Date**) dans la table SCD. **End Date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne **End Date** a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l'option **Fixed Year value** et saisir une année fictive. **Log Active Status** : Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut **true** et **false**. Cette colonne permet de repérer facilement le journal actif. **Log versions** : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.
Advanced settings	*Additional JDBC parameters*	Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.

	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type **Row Main**.	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *tDB2SCD* et *tMysqlSCD*.

tGreenplumSCD

Propriétés du tGreenplumSCD

Famille du composant	Databases/Business Intelligence/Greenplum	
Fonction	Le composant **tGreenplumSCD** reflète et traque les modifications d'une table Greenplum SCD dédiée.	
Objectif	Le **tGreenplumSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving Mode	Cochez cette case pour améliorer les performances du système.
	Source keys include Null	Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null. *Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.*
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*End date time details*	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.	

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tInformixSCD

Propriétés du tInformixSCD

Famille de composant	Databases/ Business Intelligence/ Informix	
Fonction	Le composant **tInformixSCD** reflète et traque les modifications d'une table Informix SCD dédiée.	
Objectif	Le **tInformixSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Instance	Nom de l'instance Informix à utiliser. Cette information se trouve généralement dans le fichier *SQL hosts*.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving Mode	Cochez cette case pour améliorer les performances du système.
	Source keys include Null	Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null. *Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.*
	Use Transaction	Décochez cette case lorsque la base de données est configurée en mode NO_LOG.
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*End date time details*	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type **Row Main** en entrée.
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tIngresSCD

Propriétés du tIngresSCD

Famille de composant	Databases/ Business Intelligence/Ingres	
Fonction	Le composant **tIngresSCD** reflète et traque les modifications d'une table Ingres SCD dédiée.	
Objectif	Le **tIngresSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Property type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Server	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving Mode	Cochez cette case pour améliorer les performances du système.
	Source keys include Null	Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null. *Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.*
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*End date time details*	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type **Row Main** en entrée.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tJasperOutput

Propriétés du tJasperOutput

Ce composant est étroitement lié à l'outil de reporting de Jaspersoft, iReport. Il lit et traite des données d'un flux d'entrée afin de créer un rapport à l'aide d'un modèle de rapport JRXML défini dans iReport.

Famille de composant	Business Intelligence/Jasper	
Fonction	Le composant **tJasperOutput** lit et traite des données d'un flux d'entrée afin de créer un rapport à l'aide d'un modèle de rapport JRXML défini dans iReport.	
Objectif	Ce composant vous permet d'utiliser l'outil iReport de Jaspersoft afin de créer un rapport dans des formats riches.	
Basic settings	*Jrxml file*	Fichier du modèle de rapport créé dans iReport.
	Temp path	Chemin d'accès aux fichiers temporaires.
	Destination path	Chemin d'accès au fichier final du rapport.
	File name/Stream	Nom du rapport final.
	Report type	Type du fichier du rapport final.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Permet de synchroniser le schéma du fichier d'entrée avec le schéma du fichier de sortie. La fonction Sync ne s'affiche que lorsqu'une connexion de type **Row** est connectée au composant de sortie.
	iReport	Editez la commande pour fournir le chemin d'accès au fichier d'exécution de iReport, par exemple en remplaçant *__IREPORT_PATH__* par *E:\Program Files\Jaspersoft\iReport-4.1.1\bin*, ou en saisissant le chemin d'accès complet au fichier d'exécution, par exemple *"E:\Program Files\Jaspersoft\iReport-4.1.1\bin\iReport.exe"*.
	Launch	Cliquez sur ce bouton pour lancer iReport.

Advanced settings	*Specify Locale*	Cochez cette case afin de sélectionner la langue de votre rapport dans la liste **Report Locale**. La première ligne de la liste **Report Locale** est vide. Vous pouvez cliquer dessus pour personnaliser le langage.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est étroitement lié à l'outil de reporting de Jaspersoft, iReport. Il lit et traite des données d'un flux d'entrée afin de créer un rapport à l'aide d'un modèle de rapport JRXML défini dans iReport.	

Scénario : Générer un rapport avec un modèle .jrxml

Le Job suivant lit des données d'un fichier .csv et crée un rapport .pdf basé sur un modèle de rapport .jrxml existant. Notez que le fichier de modèle doit avoir été créé via l'outil iReport de Jaspersoft à partir d'un fichier partageant le même schéma avec le fichier source .csv du Job.

Configurer votre Job

1. Glissez-déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tFileInputDelimited** et **tJasperOutput**.
2. Reliez le **tFileInputDelimited** au **tJasperOutput** à l'aide d'un lien **Row**.

Configurer le composant d'entrée

1. Double-cliquez sur le composant **tFileInputDelimited** afin d'afficher sa vue **Basic settings**.

2. Sélectionnez **Built-In** dans la liste déroulante **Property Type**.

Vous pouvez sélectionner **Repository** dans la liste **Property Type** afin de renseigner automatiquement les champs correspondants si la métadonnée a été stockée localement dans le **Repository**. Pour plus d'informations concernant les métadonnées **Metadata**, consultez le *Guide utilisateur du Studio Talend*.

3. Renseignez le champ **File name/Stream** en saisissant le chemin d'accès et le nom du fichier source, par exemple *"C:/Documents and Settings/Andy ZHANG/nom.csv"*.

4. Laissez les paramètres par défaut pour les champs **Row Separator** et **Field Separator**. Cependant, si nécessaire, vous pouvez les modifier selon vos besoins.

5. Saisissez *1* dans le champ **Header** et *0* dans le champ **Footer**. Laissez le champ **Limit** vide. Cependant, si nécessaire, vous pouvez les modifier selon vos besoins.

6. Sélectionnez **Built-In** dans la liste déroulante **Schema** puis cliquez sur **Edit schema** afin de définir la structure des données du fichier d'entrée. Dans ce cas, le fichier d'entrée contient deux colonnes : *Nom* et *Prenom*.

Configurer le composant de sortie

1. Double-cliquez sur le **tJasperOutput** pour afficher sa vue **Basic settings** et configurer ses propriétés.

2. Saisissez le chemin d'accès complet au fichier de modèle du rapport créé via l'outil iReport de Jaspersoft dans le champ **Jrxml file**. Vous pouvez également cliquer sur le bouton **[...]** afin de parcourir votre système.

   Le schéma du fichier utilisé pour créer un modèle .jrxml via iReport, doit être le même que celui du fichier source utilisé pour créer le rapport.

3. Dans le champ **Temp path**, saisissez le chemin d'accès aux fichiers temporaires générés durant l'exécution du Job. Vous pouvez également cliquez sur le bouton **[...]** afin de parcourir votre système.

4. Saisissez le chemin d'accès au rapport final généré durant l'exécution du Job, dans le champ **Destination path**. Vous pouvez également cliquer sur le bouton **[...]** afin de parcourir votre système.

5. Saisissez le nom du rapport final généré durant l'exécution du Job, dans le champ **File name/Stream**.

6. Dans le champ **Report type**, sélectionnez le format du rapport final.

7. Cliquez sur **Sync columns** afin de récupérer le schéma du composant précédent.

8. Saisissez le chemin d'accès au fichier d'exécution de iReport dans le champ **iReport**, en remplaçant par exemple *__IREPORT_PATH__* par *E:\Program Files\Jaspersoft\iReport-4.1.1\bin*. Vous pouvez cliquer sur le bouton **Launch** afin de lancer iReport.

   Cette étape n'est pas obligatoire. Cependant, elle vous permet d'accéder à l'outil iReport pour effectuer les opérations nécessaires, comme la création d'un modèle de rapport, etc.

Exécuter votre Job

1. Sauvegardez le Job.

2. Appuyez sur **F6** pour l'exécuter.

   Le fichier *out.pdf* est créé à l'emplacement spécifié dans le champ **Destination path**.

tJasperOutputExec

Propriétés du tJasperOutputExec

Ce composant est étroitement lié à l'outil de reporting de Jaspersoft, iReport. Il lit et traite des données d'un fichier afin de créer un rapport à l'aide d'un modèle de rapport .jrxml défini via iReport. Ce composant permet un gain de performance, car il est la combinaison d'un composant d'entrée et d'un composant **tJasperOutput**. L'utilisation de ces deux composants séparés permet de transformer les données avant de les utiliser pour générer un rapport, avec des sources riches et variées en entrée.

Famille de composant	Business Intelligence/ Jasper	
Fonction	Le composant **tJasperOutputExec** lit et traite des données à partir d'un fichier source pour créer un rapport à l'aide d'un modèle de rapport .jrxml défini via iReport.	
Objectif	Ce composant vous permet d'utiliser l'outil iReport de Jaspersoft afin de créer un rapport dans des formats riches. Il permet un gain de performance, puisqu'il est la combinaison d'un composant d'entrée et du composant de sortie **tJasperOutput**.	
Basic settings	*Jrxml file*	Fichier du modèle de rapport créé dans iReport
	Source file	Nom du fichier source.
	Record delimiter	Saisissez votre délimiteur d'enregistrements.
	Destination path	Chemin d'accès au fichier final du rapport.
	Use Default Output Name	Cochez cette case pour utiliser le nom par défaut du rapport généré, c'est-à-dire le nom du fichier source.
	Output Name	Nom du rapport final. Ce champ n'apparaît pas si la case **Use Default Output Name** est cochée.
	Report type	Type du fichier du rapport final.
	iReport	Editez la commande pour fournir le chemin d'accès au fichier d'exécution de iReport, par exemple en remplaçant *__IREPORT_PATH__* par *E:\Program Files\Jaspersoft \iReport-4.1.1\bin*, ou en saisissant le chemin d'accès complet au fichier d'exécution, par exemple *"E:\Program Files\Jaspersoft \iReport-4.1.1\bin\iReport.exe"*.
	Launch	Cliquez sur ce bouton pour lancer iReport.
Advanced settings	*Specify Locale*	Cochez cette case afin de sélectionner la langue de votre rapport dans la liste **Report Locale**. La première ligne de la liste **Report Locale** est vide. Vous pouvez cliquer dessus pour personnaliser le langage.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est étroitement lié à l'outil de reporting de Jaspersoft, iReport. Il lit et traite des données à partir d'un fichier source pour créer un rapport à l'aide d'un modèle de rapport .jrxml défini via iReport.

Scénario associé

Pour un scénario associé, consultez *Scénario : Générer un rapport avec un modèle .jrxml*.

tLineChart

Propriétés du tLineChart

Famille de composant	Business Intelligence/Charts	
Fonction	Le composant **tLineChart** lit des données à partir d'un flux d'entrée et transforme les données en graphique en lignes dans un fichier image au format PNG.	
Objectif	Le composant **tLineChart** génère un graphique en lignes à partir des données d'entrée afin de faciliter l'analyse technique.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Le schéma du **tLineChart** contient trois colonnes en lecture seule nommées respectivement **series** (type String), **category** (type String), et **value** (type Integer), selon un ordre prédéterminé. Les données de toutes les autres colonnes seront simplement passées au composant suivant, s'il y en a un, mais elles ne seront pas présentes dans le diagramme en barres.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Permet de synchroniser le schéma du fichier d'entrée avec le schéma du fichier de sortie. La fonction Sync ne s'affiche que lorsqu'une connexion de type **Row** est connectée au composant de sortie.
	Generated image path	Nom et chemin d'accès du fichier image de sortie.
	Chart title	Saisissez le titre du graphique en lignes à générer.
	Domain axis label et *Range axis label*	Saisissez le nom de l'axe *domain* (axe X) et le nom de l'axe *range* (axe Y).
	Plot orientation	Choisissez l'orientation qu'aura votre graphique en lignes : **VERTICAL** ou **HORIZONTAL**.
	Include legend	Saisissez *true* dans ce champ si vous souhaitez inclure une légende indiquant les lignes des séries dans des couleurs différentes ; sinon, saisissez *false*.
	Image width et *Image height*	Saisissez la largeur et la hauteur du fichier image, exprimées en pixels.
	Moving average	Cochez cette case afin d'afficher la moyenne mobile de chaque série dans votre graphique en lignes. Lorsque cette case est cochée, le champ **Period** apparaît et vous pouvez définir la période dans laquelle apparaît la moyenne mobile.
	Lower bound and *Upper bound*	Définissez quelles sont les plus hautes et les plus basses valeurs qui seront affichées sur l'axe *range*.
	Chart background et *Plot background*	Choisissez dans la liste la couleur de fond du graphique ainsi que la couleur de fond du plan du graphique (zone de l'orientation).
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant d'entrée ainsi qu'une connexion de type **Row Main**.

Scénario : Créer un graphique en lignes afin de faciliter l'analyse des tendances

Ce scénario décrit un Job simple qui lit les données d'un fichier CSV et les transforme en graphique en lignes afin de faciliter l'analyse des tendances. Le fichier d'entrée indique combien de temps (en minutes) par semaine une personne regarde différentes chaînes de télévision, sur une période de dix semaines, comme présenté ci-dessous :

```
Week;TV_A;TV_B;TV_C
1;327;286;244
2;326;285;243
3;325;283;245
4;323;282;246
5;322;285;248
6;321;288;247
7;322;291;245
8;321;292;244
9;320;293;243
10;319;294;242
```

Le fichier d'entrée ayant une structure différente de celle requise pour l'utilisation d'un **tLineChart**, un **tMap** sera utilisé dans ce scénario, afin de mettre en correspondance les données avec un fichier CSV ayant la structure requise, avant d'utiliser le **tLineChart** pour générer un fichier représentant le graphique en lignes.

Le **tMap** est généralement utilisé afin d'adapter le schéma d'entrée, conformément à la structure du schéma du composant **tLineChart**. Pour plus d'informations concernant le composant **tMap**, consultez le *Guide utilisateur du Studio Talend* ainsi que le composant *tMap*.

Déposer et relier les composants

1. A partir de la **Palette**, déposez les composants suivants dans l'espace de modélisation graphique : deux composants **tFileInputDelimited**, un **tMap**, trois composants **tFileOutputDelimited** et un **tLineChart**.
2. Connectez le premier composant **tFileInputDelimited** au **tMap** à l'aide d'une connexion **Row** > **Main**.
3. Reliez le **tMap** au premier **tFileOutputDelimited** à l'aide d'une connexion **Row** > **Main** et appelez-la *TV_A*.
4. Répétez l'étape ci-dessus : connectez le **tMap** aux deux composants **tFileOutputDelimited** restants à l'aide d'une connexion **Row** > **Main** et appelez-les respectivement *TV_B* et *TV_C*.
5. Connectez le composant **tFileInputDelimited** au **tLineChart** à l'aide d'une connexion **Row** > **Main**. Cliquez sur **Yes** pour valider la propagation du schéma à partir du composant cible.

6. Connectez le premier composant **tFileInputDelimited** au second **tFileInputDelimited** à l'aide d'une connexion **Trigger** > **OnSubjobOk**.

7. Connectez le premier **tFileDelete** au premier composant **tFileInputDelimited**, et le second **tFileInputDelimited** au second **tFileDelete** à l'aide d'une connexion **Trigger** > **OnSubjobOk**.

8. Renommez les composants comme vous le souhaitez afin qu'ils décrivent au mieux leur fonction.

Lire les données source

1. Double-cliquez sur le premier composant **tFileInputDelimited** afin d'afficher l'onglet **Basic settings**.

2. Renseignez le chemin d'accès au fichier à traiter dans le champ **File name** en parcourant votre répertoire et en sélectionnant le fichier d'entrée.

3. Spécifiez dans le champ **Header** le nombre de lignes d'en-tête, une dans ce scénario. Laissez les autres paramètres tels qu'ils sont.

4. Cliquez sur **Edit schema** pour décrire la structure des données du fichier d'entrée. Dans ce scénario, le schéma d'entrée est composé de quatre colonnes : *Week*, *Mins_TVA*, *Mins_TVB*, et *Mins_TVC*. Une fois le nom des colonnes et les types de données définis, cliquez sur **OK** pour fermer la boîte de dialogue du schéma.

Adapter les données source au schéma tLineChart

1. Double-cliquez sur le **tMap** pour ouvrir l'éditeur **Map Editor**.

   Le panneau de sortie contient une table d'entrée, ici *row1*, et le panneau de sortie contient trois tables vides nommées *TV_A*, *TV_B* et *TV_C*.

2. Utilisez l'éditeur **Schema editor** pour ajouter trois colonnes à chacune des tables de sortie : *series* (type String), *x* (type Integer) et *y* (type Integer).

3. Dans le champ **Expression** de chaque table de sortie, saisissez les noms des *series*, comme indiqué ci-dessus. Ces noms apparaîtront dans la légende du graphique en lignes.

4. Déposez la colonne *Week* de la table d'entrée dans la colonne *x* de chaque table de sortie.

5. Déposez la colonne *Mins_TVA* de la table d'entrée dans la colonne *y* de la table *TV_A*.

6. Déposez la colonne *Mins_TVB* de la table d'entrée dans la colonne *y* de la table *TV_B*.

7. Déposez la colonne *Mins_TVC* de la table d'entrée dans la colonne *y* de la table *Mins_TV_C*.

8. Cliquez sur **OK** pour sauvegarder le mapping, fermer l'éditeur et propager le schéma de sortie au composant de sortie.

Générer le fichier d'entrée temporaire

1. Double-cliquez sur le premier composant **tFileOutputDelimited** pour afficher l'onglet **Basic settings**.

2. Dans le champ **File Name**, définissez un fichier CSV dans lequel les flux de données mappés seront envoyés. Dans cet exemple, nommez le fichier de sortie à créer *Temp.csv*. Ce fichier sera utilisé comme fichier d'entrée pour le composant **tLineChart**.

3. Cochez la case **Append**.

4. Définissez les paramètres des deux autres composants **tFileOutputDelimited** exactement de la même manière que dans le premier **tFileOutputDelimited.**

Notez que l'ordre des flux de données sortant du composant **tMap** n'est pas nécessairement l'ordre dans lequel les données seront écrites dans le composant cible. Pour vous assurer que le fichier est correctement généré, supprimez, s'il existe, le fichier portant le même nom avant d'exécuter le Job. Vérifiez que la case **Append** est cochée dans tous les composants **tFileOutputDelimited** afin que tous les flux de données mappés aillent dans le même fichier, sans écraser les données existantes.

Configurer le tLineChart

1. Double-cliquez sur le second composant **tFileInputDelimited** afin d'afficher son onglet **Basic settings**.

2. Saisissez, dans le champ **File name**, le chemin d'accès au fichier, ainsi que son nom, le même que celui défini dans chaque composant **tFileOutputDelimited**. Dans cet exemple, le fichier d'entrée du **tLineChart** est *Temp.csv*.

3. Double-cliquez sur le composant **tLineChart** pour afficher l'onglet **Basic settings**.

4. Cliquez sur **Edit schema** afin d'ouvrir la boîte de dialogue du schéma.

5. Le schéma d'entrée et de sortie doivent être synchronisés. Si besoin est, copiez toutes les colonnes du schéma de sortie dans le schéma d'entrée en cliquant sur la double-flèche pointant vers la gauche. Cliquez sur **OK** pour fermer la boîte de dialogue du schéma.

6. Dans le champ **Generated image path**, définissez le chemin d'accès du fichier image à générer.

7. Dans le champ **Chart title**, saisissez un titre pour le graphique en lignes. Dans ce scénario, son titre est `Average Weekly Viewing (per person)`.

8. Saisissez un nom pour les axes (X) et (Y). Dans ce scénario, le nom des axes est respectivement `Week` et `Minutes`.

9. Définissez comme vous le souhaitez la taille de l'image (dans les champs **Image width** pour la largeur et **Image height** pour la hauteur), la période dans laquelle est calculée la moyenne mobile (dans le champ **Period** une fois la case **Moving average** cochée), les limites inférieures et supérieures (respectivement dans les champs **Lower bound** et **Upper bound**), ainsi que la couleur de fond du graphique et du plan (respectivement dans les champs **Chart background** et **Plot background**).

   Dans ce scénario, définissez la taille de l'image à *450 by 450*, paramétrez le champ **Lower bound** à *210* et **Upper bound** à *340*, sélectionnez *Light_gray* dans la liste **Chart background** et laissez les autres paramètres tels qu'ils sont.

Supprimer le fichier temporaire

1. Double-cliquez sur le premier composant **tFileDelete** pour afficher sa vue **Basic settings**.

2. Dans le champ **File name**, renseignez le chemin d'accès du fichier d'entrée et décochez la case **Fail on error** pour permettre au Job principal d'être exécuté si le fichier à supprimer n'existe pas.

3. Spécifiez le même chemin d'accès que dans l'autre composant **tFileDelete**.

Exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** pour l'exécuter.

Un graphique en lignes est généré selon les critères définis. Il affiche une comparaison du temps de visionnage moyen et des évolutions dans le temps correspondant aux différentes chaînes de télévision.

Average Weekly Viewing (per person)
Minutes
340
330
320
310
300
290
280
270
260
250
240
230
220
210
1
2
3
4
5
6
7
8
9
10
Week
TV A
TV B
TV C

tMondrianInput

Propriétés du tMondrianInput

Famille du composant	Business Intelligence/OLAP Cube	
Fonction	Le composant **tMondrianInput** lit des données à partir de bases relationnelles et produit des groupes de données multidimensionnels reposant sur une requête MDX.	
Objectif	Le **tMondrianInput** exécute une expression multi-dimensionnelle (MDX) correspondant à la structure du groupe de données et au schéma défini. Puis il passe les données multidimensionnelles obtenues au composant suivant à l'aide d'un lien de type **Row Main**.	
Basic settings	*Mondrian Version*	Sélectionnez la version de Mondrian que vous utilisez.
	DB type	Sélectionnez le type de relation approprié pour relier les bases de données.
	Property type	Peut être Built-in ou Repository.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Datasource	Chemin d'accès et nom du fichier contenant les données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Consultez également le *Guide utilisateur du Studio Talend*.
	Catalog	Chemin d'accès au catalogue (structure de l'entrepôt de données).

	MDX Query	Saisissez votre requête MDX en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma et dans la structure de l'entrepôt de données.
	Encoding	Sélectionnez le type d'encodage à partir de la liste ou choisissez l'option **Custom** pour le définir manuellement. Ce champ doit obligatoirement être renseigné.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **Flow** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités des requêtes MDX pour des groupes de données multidimensionnels.	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario : Tables en jointure croisée

Le présent Job permet d'extraire des groupes de données multidimensionnels à partir de tables de bases de données relationnelles stockées dans une base MySQL. Les données sont récupérées à l'aide d'une expression multidimensionnelle (requête MDX). Il est clair que vous devez connaître la structure de vos données, ou du moins disposer d'une description de la structure (catalogue) comme référence du groupe de données à récupérer dans ces différentes dimensions.

Configurer le Job

1. A partir de la **Palette**, cliquez-déposez un composant **tMondrianInput** (famille **Business Intelligence**) ainsi qu'un composant de sortie **tLogRow** (famille **Logs & Errors**).

2. Reliez le **tMondrianInput** au composant de sortie à l'aide d'un lien de type **Row Main**.

Configurer la connexion à la base de données

- Sélectionnez le composant **tMondrianInput** puis cliquez sur la vue **Component**.

1. Dans le champ **DB Type**, sélectionnez la base de données relationnelle à utiliser avec Mondrian.

2. Sélectionnez l'entrée du **Repository** correspondante dans le champ **Property type**, dans le cas où vous stockez les détails de connexion à vos bases de données de manière centralisée. Dans cet exemple, les propriétés sont en mode **Built-in**.

3. Remplissez les détails de connexion à votre base de données : **Host**, **Port**, **Database**, **User Name** et **Password**.

4. Dans le champ **Schema**, sélectionnez le schéma correspondant à partir du **Repository** dans le cas où il est stocké de manière centralisée. Dans cet exemple, le schéma doit reste à définir (mode **Built-in**).

Configurer la requête sur la base de données

1. La base de données relationnelle que l'on veut interroger contient cinq colonnes : *media*, *drink*, *unit_sales*, *store_cost* et *store_sales*.

2. La requête doit permettre de récupérer le contenu de *unit_sales*, *store_cost* et de *store_sales* pour les différentes valeur de *media* / *drink* en utilisant une requête MDX comme dans l'exemple ci-dessous :

CrossJoin Example 1

The current slicer is **1997**.

		Unit Sales	Store Cost	Store Sales
Radio	Alcoholic Beverages	75	70.40	168.62
	Beverages	97	75.70	186.03
	Dairy	54	36.75	89.03
TV	Alcoholic Beverages	76	70.99	182.38
	Beverages	188	167.00	419.14
	Dairy	68	45.19	119.55
Sunday Paper	Alcoholic Beverages	148	128.97	316.88
	Beverages	197	161.81	399.58
	Dairy	85	54.75	140.27
Street Handout	Alcoholic Beverages	158	121.14	294.55
	Beverages	270	201.28	520.55
	Dairy	84	50.26	128.32

1. A partir de l'onglet **Basic settings** du composant **tMondrianInput**, renseignez le chemin d'accès au catalogue de l'entrepôt de données dans le champ **Catalog**. Ce catalogue décrit la structure de l'entrepôt.

2. Entrez ensuite la requête MDX de la manière suivante :

```
"select
   {[Measures].[Unit Sales], [Measures].[Store Cost], [Measures].[Store
Sales]} on columns,
   CrossJoin(
     { [Promotion Media].[All Media].[Radio],
       [Promotion Media].[All Media].[TV],
       [Promotion Media].[All Media].[Sunday Paper],
       [Promotion Media].[All Media].[Street Handout] },
     [Product].[All Products].[Drink].children) on rows
 from Sales
 where ([Time].[1997])"
```

3. Sélectionnez enfin le type d'encodage à partir de la liste du champ **Encoding Type**.

Exécuter le Job

1. Sélectionnez le composant **tLogRow** puis cochez l'option **Print header** pour afficher le nom des colonnes sur la console.

2. Enregistrez le Job puis appuyez sur **F6** pour l'exécuter.

```
Starting job Mondrian at 15:31 08/01/2008.
media   drink   unit sales  store cost  store sales
Promotion Media].[All Media].[Radio]|[Product].[All Products].[Drink].[Alcoholic
everages]|75.0|70.3977|168.62
Promotion Media].[All Media].[Radio]|[Product].[All Products].[Drink].[Beverages]|97.0|75.7016|186.03
Promotion Media].[All Media].[Radio]|[Product].[All Products].[Drink].[Dairy]|54.0|36.7474|89.03
Promotion Media].[All Media].[TV]|[Product].[All Products].[Drink].[Alcoholic
everages]|76.0|70.9895|182.38
Promotion Media].[All Media].[TV]|[Product].[All Products].[Drink].[Beverages]|188.0|167.0025|419.14
Promotion Media].[All Media].[TV]|[Product].[All Products].[Drink].[Dairy]|68.0|45.1909|119.55
Promotion Media].[All Media].[Sunday Paper]|[Product].[All Products].[Drink].[Alcoholic
everages]|148.0|128.9736|316.88
Promotion Media].[All Media].[Sunday Paper]|[Product].[All Products].[Drink].
Beverages]|197.0|161.8067|399.58
Promotion Media].[All Media].[Sunday Paper]|[Product].[All Products].[Drink].
Dairy]|85.0|54.746|140.27
Promotion Media].[All Media].[Street Handout]|[Product].[All Products].[Drink].[Alcoholic
everages]|158.0|121.1394|294.55
Promotion Media].[All Media].[Street Handout]|[Product].[All Products].[Drink].
Beverages]|270.0|201.2816|520.55
Promotion Media].[All Media].[Street Handout]|[Product].[All Products].[Drink].
Dairy]|84.0|50.2604|128.32
Job Mondrian ended at 15:31 08/01/2008. [exit code=0]
```

La console affiche le résultat de *unit_sales*, *store_cost* et *store_sales* pour chaque type de *Drink* (*Beverages*, *Dairy*, *Alcoholic beverages*) croisé avec chaque type de *media* (*TV*, *Sunday Paper*, *Street handout*) comme illustré ci-dessus.

tMSSqlSCD

Propriétés du tMSSqlSCD

Famille de composant	Databases/Business Intelligence/MSSQL Server	
Fonction	Le composant **tMSSqlSCD** reflète et traque les modifications d'une table SCD MSSQL dédiée.	
Objectif	Le **tMSSqlSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Server	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Schema	Nom du schéma de la base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations concernant les méthodes de création, consultez *Clés SCD*.
	Use memory saving Mode	Cochez cette case pour améliorer les performances du système.
	Source keys include Null	Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null. *Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.*
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	End date time details	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tMysqlSCD

Propriétés du tMysqlSCD

Famille de composant	Databases/Business Intelligence/ MySQL	
Fonction	Le composant **tMysqlSCD** reflète et traque les modifications d'une table SCD MySQL dédiée.	
Objectif	Le **tMysqlSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version de Mysql que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.

		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving mode	Cochez cette case pour améliorer les performances du système.
	Source keys include Null	Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null. *Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.*
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Additional JDBC Parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	End date time details	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type **Row** > **Main** en entrée.

Méthodologie de gestion du SCD

Lors de dimensions à évolution lente ou Slowly Changing Dimensions (SCD), les données changent lentement. L'éditeur SCD permet de construire de manière simple le flux de sortie des données SCD. Dans l'éditeur SCD, vous pouvez relier les colonnes, sélectionner la clé de substitution (Surrogate key) et paramétrer les attributs des colonnes modifiées en combinant les différents types de modifications SCD.

La figure ci-dessous représente l'éditeur SCD.

Clés SCD

Vous devez sélectionner une ou plusieurs colonnes qui serviront de clés permettant d'assurer l'unicité des données entrantes.

Vous devez aussi sélectionner une colonne sur laquelle positionner une clé de substitution (surrogate key) dans la table SCD et la relier à une des colonnes d'entrée de la table source. La valeur de la clé de substitution permet de relier les enregistrements de la table source aux enregistrements de la table SCD. L'éditeur utilise ce mapping pour localiser l'enregistrement dans la table SCD et pour déterminer si un enregistrement est nouveau ou s'il a été modifié. La clé de substitution est généralement la clé primaire de la table source, mais elle peut aussi correspondre à une autre clé à partir du moment où elle permet d'identifier de façon unique un enregistrement et où ses valeurs ne changent pas.

Source keys : Glissez une ou plusieurs colonnes du panneau **Unused** vers le panneau **Source keys**. Ces colonnes seront utilisées en tant que clé(s) assurant l'unicité des données entrantes.

Surrogate keys : Paramétrez la colonne dans laquelle la clé de substitution générée sera stockée. Une clé de substitution peut être générée en fonction de la méthode sélectionnée dans la liste **Creation**.

Creation : Sélectionnez une des méthodes suivantes permettant de générer la clé :

- **Auto increment** : la clé est auto-incrémentée.
- **Input field** : la clé est fournie par un champ d'entrée.

  Lorsque cette méthode est sélectionnée, vous pouvez glisser le champ correspondant du panneau **Unused** vers le champ **complement**.
- **Routine** : à partir du champ **complement**, vous pouvez appuyer sur **Ctrl+ Espace** afin d'afficher la liste d'auto-complétion et de sélectionner la routine appropriée.
- **Table max +1** : la valeur maximum de la table SCD est incrémentée pour créer une clé de substitution.
- **DB Sequence** : à partir du champ **complement**, vous pouvez saisir le nom de la séquence de la base de données (**DB Sequence**) existante qui va incrémenter automatiquement la colonne indiquée dans le champ **name**.

L'option **DB Sequence** est disponible uniquement depuis l'éditeur **SCD** du composant **tOracleSCD**.

Types de SCD

Il existe quatre types de dimensions à évolution lente (Slowly Changing Dimensions) : du **Type 0** au **Type 3**. Vous pouvez, d'un simple glisser-déposer, appliquer n'importe quel type de SCD à n'importe quelle colonne de la table source.

- **Type 0** : ce type de SCD n'est pas beaucoup utilisé. Certaines données dimensionnelles peuvent être écrasées et d'autres peuvent rester inchangées au cours du temps. Ce type de SCD convient lorsque aucun effort n'a été mis en place pour gérer les dimensions à caractère évolutif.
- **Type 1** : aucun historique n'est stocké dans la base de données. Les nouvelles données écrase les anciennes. Utilisez ce type si vous n'avez pas besoin de traquer les changements. Ce type de SCD convient lorsque vous corrigez certaines fautes, par exemple l'orthographe des mots.
- **Type2** : l'intégralité de l'historique est stockée dans la base de données. Ce type de SCD traque les données d'historique en enregistrant un nouvel enregistrement dans la table de dimension avec une nouvelle clé à chaque fois qu'un changement est effectué. Ce type de SCD convient lorsque l'on traque les mises à jour, par exemple.

Le principe du SCD **Type 2** réside dans le fait qu'un nouvel enregistrement est ajouté à la table SCD lorsqu'un changement est détecté dans les colonnes sélectionnées. Notez que bien que plusieurs changements peuvent être effectués au même enregistrement sur plusieurs colonnes paramétrées en **Type 2**, une seule ligne traquant ces changements est ajoutée à la table SCD.

Le schéma du SCD Type 2 devrait inclure des colonnes spécifiques au SCD contenant les informations de log standard, notamment :

-**start** : ajoute une colonne au schéma de la table SCD contenant la date de début d'un enregistrement. Vous pouvez sélectionner une des colonnes du schéma d'entrée comme date de départ pour la table SCD.

-**end** : ajoute une colonne au schéma de la table SCD contenant la date de fin d'un enregistrement. Lorsque l'enregistrement est en cours, la date de fin est de valeur **NULL** , sinon vous pouvez utiliser une année fixe en sélectionnant **Fixed Year Value** dans la liste et renseignez l'année fictive dans la cellule d'à côté pour éviter d'avoir une valeur nulle dans le champ **end**.

-**version** : ajoute une colonne au schéma de la table SCD contenant le numéro de version de l'enregistrement.

-**active** : ajoute une colonne au schéma de la table SCD contenant les statuts **true** ou **false**. Cette colonne permet de repérer facilement les enregistrements actifs.

- **Type 3** : seules les informations sur l'ancienne valeur d'une dimension est écrite dans la base de données. Ce type de SCD traque les changements en utilisant des colonnes séparées. Ce type de SCD convient lorsque l'on traque les valeurs précédentes d'une colonne qui change.

Scénario : Traquer des modifications avec les Slowly Changing Dimensions de type 0 à 3

Ce scénario décrit un Job traquant les modifications effectuées dans quatre des colonnes du fichier délimité source, écrit ces modifications ainsi que l'historique des modifications dans une table SCD et affiche les informations d'erreurs dans la console **Run**.

Le fichier délimité source contient différentes informations personnelles : des prénoms dans la colonne *firstname*, des noms dans la colonne *lastname*, des adresses dans la colonne *address*, des villes dans la colonne *city*, l'entreprise de la personne dans la colonne *company*, son âge dans la colonne *age* et son statut marital dans la colonne *status*. Une colonne *id* permet d'assurer l'unicité des données.

	A	B	C	D	E	F
1	id;firstname;lastname;address;city;company;age;status					
2	1;Janet;Anderson;511 Maple Ave. Apt. 1B;Agoura Hills;Adecco;30;married					
3	2;Harold;Smith;662 Lyons Circle;Alameda;Adidas;33;married					
4	3;Adam;Brown;220 Vine Ave.;Albany;Bridgestone;28;single					
5	4;Martin;Clinton;770 Exmoor Rd.;Alhambra;Cutco;25;single					
6	5;Sue;White;1486 Oakwood;Covina;Black&White;35;married					

Les changements de statut marital doivent écraser les anciens statuts enregistrés. Ce type de changement correspond à un changement SCD **Type 1**.

Un nouvel enregistrement dans la table dimensionnelle avec une clé différente sera inséré à chaque fois qu'une personne change d'entreprise. Ce type de changement correspond à un changement SCD **Type 2**.

Vous allez récupérer l'ancienne ville et l'ancienne adresse d'une personne. Ce type de changement correspond à un changement SCD **Type 3**.

Pour réaliser ce type de scénario, il est préférable de le séparer en trois étapes principales : la définition du flux principal du Job, le paramétrage de l'éditeur SCD et la création de la table de modification dans la base de données.

Définir le flux principal du Job

1. Cliquez et déposez les composants suivants dans l'éditeur graphique : **tMysqlConnection**, **tFileInputDelimited**, **tMysqlSCD**, **tMysqlCommit** ainsi que deux composants **tLogRow**.

2. Connectez les composants **tFileInputDelimited**, le premier **tLogRow** et le **tMysqlSCD** à l'aide d'un lien **Row Main**. Ce sous-job correspond au flux principal du Job.

3. Connectez le **tMysqlConnection** au **tFileInputDelimited** et le **tMysqlSCD** au **tMysqlCommit** à l'aide de liens **OnComponentOk**.

4. Connectez le **tMysqlSCD** au second **tLogRow** à l'aide d'un lien **Row Reject**. Les colonnes *errorCode* et *errorMessage* sont ajoutées au schéma. Cette connexion collecte les informations d'erreurs.

Configurer la connexion à la base de données et le composant d'entrée

1. Dans l'éditeur graphique, double-cliquez sur le composant **tMysqlConnection** pour afficher ses propriétés de base dans l'onglet **Basic settings** et renseigner manuellement les informations de connexion. Lorsque plusieurs composants Database sont utilisés, le composant **tMysqlConnection** devrait être utilisé pour éviter de renseigner plusieurs fois les mêmes informations de connexion à la base de données.

   Si vous avez déjà stocké les informations de connexion dans le **Repository**, glissez l'élément **Metadata** souhaité dans l'éditeur graphique et les informations de connexion à la base de données apparaîtront automatiquement dans les champs correspondants. Pour plus d'informations sur les métadonnées, consultez le *Guide utilisateur du Studio Talend*.

   Dans ce scénario, l'objectif est de se connecter à la table SCD dans laquelle toutes les modifications apportées au fichier délimité seront traquées.

2. Dans l'éditeur graphique, double-cliquez sur le composant **tFileInputDelimited** pour afficher la vue **Basic settings**.

3. Cliquez sur le bouton **[...]** à côté du champ **File Name** pour sélectionner le chemin d'accès au fichier délimité *dataset.csv* source contenant les informations personnelles.

4. Dans les champs **Row** et **Field Separator**, renseignez respectivement les séparateurs de lignes et de champs utilisés dans le fichier source.

   Les champs **File Name**, **Row separator** et **Field separators** sont obligatoires.

5. Si nécessaire, renseignez les champs **Header**, **Footer** et **Limit**.

   Dans ce scénario, saisissez *1* dans le champ **Header** puisque le fichier source contient un en-tête. Laissez le champ **Footer** à *0* puisque le fichier source ne contient pas de pied de page et ne renseignez pas le champ **Limit** déterminant le nombre de lignes devant être traitées.

6. Cliquez sur **Edit schema** pour décrire la structure des données du fichier délimité source.

   Dans ce scénario, le schéma source est composé de huit colonnes : *id*,*firstname*, *lastname*, *address*, *city*, *company*, *age* et *status*.

7. Dans l'onglet **Basic settings** du premier composant **tLogRow**, sélectionnez le mode **Table** afin que le contenu du fichier source, ainsi que les attributs variants apparaissent dans les cellules du tableau dans la console avant d'être traitées par le composant SCD.

Configurer tMysqlSCD et tMysqlCommit

1. Dans l'éditeur graphique, double-cliquez sur le composant **tMysqlSCD** pour paramétrer ses propriétés.

2. Dans l'onglet **Basic settings**, cochez la case **Use an existing connection** pour réutiliser les informations de connexion définies dans les propriétés du composant **tMysqlConnection**.

3. Dans le champ **Table**, saisissez le nom de la table SCD dans laquelle les changements seront stockés.

4. Si nécessaire, cliquez sur **Sync columns** pour récupérer toutes les colonnes du fichier délimité.

5. Dans l'éditeur graphique, double-cliquez sur le composant **tMysqlCommit** pour paramétrer ses propriétés.

6. Dans la liste **Component list**, sélectionnez la connexion adéquate s'il y a plusieurs composants de connexion dans le Job.

7. Dans l'onglet **Basic settings** du second composant **tLogRow**, sélectionnez le mode **Table** afin que le contenu du fichier source, ainsi que les attributs variants apparaissent dans les cellules d'un tableau.

Paramétrer l'éditeur SCD

1. Dans l'onglet **Basic settings** du composant **tMysqlSCD**, cliquez sur le bouton **[...]** à côté du champ **SCD Editor** pour ouvrir l'éditeur **SCD editor** et construire le flux de données en sortie du SCD.

   Toutes les colonnes du fichier délimité source, provenant du composant précédent et n'ayant pas encore été utilisées, apparaissent dans le panneau **Unused** de l'éditeur **SCD**. Tous les autres panneaux de l'éditeur sont vides.

2. A partir de la liste **Unused**, glissez la colonne *id* vers le panneau **Source keys** pour être utilisée comme clé permettant d'assurer l'unicité des données entrantes.

3. Dans le panneau **Surrogate keys**, donnez un nom à la clé de substitution (surrogate key) dans le champ **Name**, **SK1** pour ce scénario.

4. Dans la liste **Creation**, sélectionnez la méthode à utiliser pour générer la clé de substitution, **Auto-increment** pour ce scénario.

5. Dans la liste **Unused**, glissez les colonnes *firstname* et *lastname* vers le panneau **Type 0**, les changements effectués dans ces colonnes ne sont pas pertinents dans cet exemple.

6. Glissez la colonne *status* vers le panneau **Type 1**. L'ancienne valeur sera écrasée par la nouvelle.

7. Glissez la colonne *company* vers le panneau **Type 2**. A chaque fois qu'une personne change d'entreprise, un nouvel enregistrement sera inséré dans la table dimensionnelle avec une nouvelle clé.

   Dans la zone **Versioning** :

   -Paramétrez les colonnes **start** et **end** de votre table SCD contenant respectivement les dates de début et de fin. La date de fin est nulle pour les enregistrements en cours jusqu'à ce qu'un changement soit détecté. A ce moment là, la date de fin est renseignée et un nouvel enregistrement est ajouté sans date de fin.

   Dans ce scénario, sélectionnez **Fixed Year Value** dans la colonne **end** et saisissez une année fictive pour éviter d'avoir une valeur nulle dans le champ de date de fin.

   -cochez le case **version** pour conserver le numéro de version de l'enregistrement.

   -cochez la case **active** pour ajouter la colonne **Status** contenant les statuts **True** ou **False** des enregistrements. **True** correspond à l'enregistrement en cours et **False** correspond à l'enregistrement modifié.

8. Glissez les colonnes *address* et *city* vers le panneau **Type 3** pour ne récupérer que les informations sur les anciennes valeurs des adresses et des villes.

Pour plus d'informations sur les types de SCD, consultez *Méthodologie de gestion du SCD*.

9. Cliquez sur **OK** pour valider votre configuration et fermer l'éditeur **SCD**.

Créer la table SCD

1. Cliquez sur **Edit schema** pour afficher le flux de données en entrée et en sortie. Le schéma de sortie du SCD devrait inclure les colonnes spécifiques au SCD contenant les informations de log standard définies dans l'éditeur **SCD editor**.

Si vous modifiez la définition du schéma d'entrée, vous devez vérifier et reconfigurer si nécessaire la définition du flux de sortie dans l'éditeur **SCD**, afin de s'assurer que la structure des données de sortie est correctement actualisée.

2. Double-cliquez sur le composant **tMysqlSCD** pour afficher sa vue **Basic settings** et sélectionnez **Create table if not exists** à partir de la liste **Action on table** pour éviter d'avoir à créer et définir manuellement la table SCD.

Exécuter le Job

Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

La console **Run** affiche le contenu du fichier délimité d'entrée et votre table SCD est créée dans votre base de données, contenant l'ensemble de données initial.

SK1	address	city	company	firstName	id	lastName	previous_adc	previous_city	scd_ac	scd_end	scd_st	scd_ve	status
1	511 Maple Ave. Apt. 1B	Agoura Hills	Adecco	Janet	1	Anderson	(NULL)	(NULL)	□	2012-01-	2011-0	1	married
2	662 Lyons Circle	Alameda	Adidas	Harold	2	Smith	(NULL)	(NULL)	□	2012-01-	2011-0	1	married
3	220 Vine Ave.	Albany	Bridgestone	Adam	3	Brown	(NULL)	(NULL)	□	2012-01-	2011-0	1	single
4	770 Exmoor Rd.	Alhambra	Cutco	Martin	4	Clinton	(NULL)	(NULL)	□	2012-01-	2011-0	1	single
5	1486 Oakwood	Covina	Black&White	Sue	5	White	(NULL)	(NULL)	□	2012-01-	2011-0	1	married
JLL)	(NULL)	(NULL)	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Janet divorce, déménage à *Adelanto* au *355 Golf Rd* et travaille maintenant chez *Greenwood*.

Adam s'est marié, a déménagé à *Belmont* au *2505 Alisson ct.* et travaille chez *Scoop*.

Martin travaille désormais chez *Phillips and Brothers*.

Mettez à jour le fichier délimité avec ces informations et appuyez sur **F6** pour exécuter le Job.

La console affiche les changements apportés aux informations personnelles et la table SCD affiche l'historique des changements valides, ainsi que leur statut et leur numéro de version. Le nom de l'entreprise de Martin dépasse la longueur définie pour la colonne *company* dans le schéma. Ce changement est donc dirigé vers le flux de rejets, et non sauvegardé dans la table SCD.

SK1	address	city	company	firstName	id	lastName	previous_add	previous_city	scd_ac	scd_end	scd_st	scd_ve	status
1	511 Maple Ave. Apt. 1B	Agoura Hills	Adecco	Janet	1	Anderson	(NULL)	(NULL)	(Binary/	2011-02-	2011-0	1	single
2	662 Lyons Circle	Alameda	Adidas	Harold	2	Smith	(NULL)	(NULL)	□	2012-01-	2011-0	1	married
3	220 Vine Ave.	Albany	Bridgestone	Adam	3	Brown	(NULL)	(NULL)	(Binary/	2011-02-	2011-0	1	married
4	770 Exmoor Rd.	Alhambra	Cutco	Martin	4	Clinton	(NULL)	(NULL)	(Binary/	2011-02-	2011-0	1	single
5	1486 Oakwood	Covina	Black&White	Sue	5	White	(NULL)	(NULL)	□	2012-01-	2011-0	1	married
6	355 Golf Rd.	Adelanto	Greenwood	Janet	1	Anderson	511 Maple Av	Agoura Hills	□	2012-01-	2011-0	2	single
7	2505 Alisson ct	Belmont	Scoop	Adam	3	Brown	220 Vine Ave.	Albany	□	2012-01-	2011-0	2	married
JLL)	(NULL)	(NULL)	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

tMysqlSCDELT

Propriétés du tMysqlSCDELT

Famille de composant	Databases/Business Intelligence/MySQL	
Fonction	Le composant **tMysqlSCDELT** reflète et traque les modifications d'une table MySQL SCD dédiée.	
Objectif	Le **tMysqlSCDELT** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table MySQL SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez la version de Mysql que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Source table	Nom de la table contenant les données à filtrer.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois pour que l'opération d'insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.

		Clear a table : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Source Keys	Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l'unicité des données entrantes.
	Use SCD Type 1 fields	Utilisez le type 1 si vous n'avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.
	Use SCD Type 2 fields	Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications. **Start date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l'une des colonnes d'entrée du schéma comme date de départ (Start Date) dans la table SCD. **End Date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne **End Date** a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l'option **Fixed Year value** et saisir une année fictive. **Log Active Status** : Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut **true** et **false**. Cette colonne permet de repérer facilement le journal actif. **Log versions** : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.
Advanced settings	*Debug mode*	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**.	

	Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type **Row Main**.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*, ainsi que *Scénario : Traquer des modifications avec les Slowly Changing Dimensions de type 0 à 3*.

tNetezzaSCD

Propriétés du tNetezzaSCD

Famille de composant	Databases/ Netezza	
Fonction	Le composant **tNetezzaSCD** reflète et traque les modifications d'une table SCD Netezza dédiée.	
Objectif	Le **tNetezzaSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Create a table** : créer une nouvelle table. **Create table if not exists** : si nécessaire, créer une table qui n'existe pas encore.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving Mode	Cochez cette case afin d'optimiser les performances du système.
	Source keys include Null	Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null. *Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.*
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	End date time details	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée et un composant de sortie.
Limitation	Le fichier *nzjdbc.jar* doit être installé séparément. Pour plus d'informations, consultez la section concernant les modules externes dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tOracleSCD

Propriétés du tOracleSCD

Famille de composant	Databases/Business Intelligence/Oracle	
Fonction	Le composant **tOracleSCD** reflète et traque les modifications d'une table SCD Oracle dédiée.	
Objectif	Le **tOracleSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Liste déroulante des pilotes disponibles: **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle. **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante. **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.

	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Create a table** : créer une nouvelle table. **Create table if not exists** : si nécessaire, créer une table qui n'existe pas encore.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving Mode	Cochez cette case pour améliorer les performances du système.
	Source keys include Null	Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null. *Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.*
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et d'ignorer les lignes en erreur.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	End date time details	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant.	

	La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tOracleSCDELT

Propriétés du tOracleSCDELT

Famille de composant	Databases/ Business Intelligence/ Oracle	
Fonction	Le composant **tOracleSCDELT** reflète et traque les modifications d'une table Oracle SCD dédiée.	
Objectif	Le **tOracleSCDELT** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table Oracle SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Liste déroulante des pilotes disponibles: **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle. **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante. **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.

	Source table	Nom de la table contenant les données à filtrer.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois pour que l'opération d'insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Surrogate Key	Sélectionnez dans la liste une colonne à utiliser comme clé de substitution.
	Creation	Sélectionnez la méthode à utiliser pour générer la clé de substitution. Pour plus d'informations concernant les méthodes de création, consultez *Clés SCD*.
	Source Keys	Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l'unicité des données entrantes.
	Source fields value include Null	Cochez cette case pour autoriser les valeurs Nulls dans les colonnes source. Les colonnes source font référence aux champs définis dans les tables **SCD type 1 fields** et **SCD type 2 fields**.
	Use SCD Type 1 fields	Utilisez le type 1 si vous n'avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.
	Use SCD Type 2 fields	Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications. **Start date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l'une des colonnes d'entrée du schéma comme date de départ (Start Date) dans la table SCD. **End Date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne **End Date** a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l'option **Fixed Year value** et saisir une année fictive.

		Log Active Status : Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut **true** et **false** . Cette colonne permet de repérer facilement le journal actif. **Log versions** : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type **Row Main**.	

Scénario associé

Pour un scénario associé, consultez *tOracleSCD*, ainsi que *tMysqlSCD*.

tPaloCheckElements

Propriétés du tPaloCheckElements

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloCheckElements** vérifie l'existence d'éléments du flux de données d'entrée dans un cube donné.	
Objectif	Ce composant peut être utilisé avec le **tPaloOutputMulti**. Il vérifie si les éléments du flux d'entrée existent dans le cube donné, avant de les écrire. Il peut également définir une valeur par défaut qui peut être utilisée si les éléments n'existent pas.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données dans laquelle écrire les données.
	Cube	Saisissez le nom du cube dans lequel les données doivent être écrites.
	On element error	Sélectionnez l'action à effectuer lorsqu'un élément n'existe pas. - **Reject row** : la ligne correspondante sera rejetée et placée dans le flux de rejet. - **Use default** : la valeur par défaut définie sera utilisée. - **Stop** : le processus complet sera interrompu.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Un tableau vous permet de définir les éléments à vérifier. - **Column** : affiche la (les) colonne(s) du schéma d'entrée. Cette colonne est automatiquement remplie une fois qu'un schéma d'entrée est saisi ou créé. - **Element type** : sélectionnez le type d'élément de la colonne d'entrée. Seule une colonne peut être définie en tant que **Measure**. - **Default** : saisissez la valeur par défaut qui sera utilisée si l'option **Use default** a été sélectionnée dans le champ **On element error**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant requiert un composant d'entrée.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main, Rejects **Trigger** : Run if, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Rejects Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant fonctionne uniquement avec des cubes Palo de type "Normal". Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *Scénario 2 : Refuser un flux de données d'entrée lorsque les éléments à écrire n'existent pas dans un cube donné*.

tPaloClose

Propriétés du tPaloClose

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Ce composant ferme une connexion active au serveur Palo.	
Objectif	Ce composant est utilisé pour désactiver une connexion à un serveur Palo afin de libérer des ressources occupées.	
Basic settings	*Component List*	Sélectionnez un composant **tPaloConnection** dans la liste déroulante si plusieurs connexions sont utilisées dans le Job.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les données au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Palo, notamment avec le **tPaloConnection**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	n/a	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tPaloConnection

Propriétés du tPaloConnection

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloConnection** ouvre une connexion vers un serveur Palo et la laisse ouverte durant le temps de traitement. Chaque autre composant Palo du processus peut utiliser cette connexion.	
Objectif	Ce composant permet le partage d'une connexion à un serveur Palo durant le processus complet.	
Basic settings	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé avec les composants Palo afin de partager une connexion à un serveur Palo.	
Connections		Liens de sortie (de ce composant à un autre) : **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus	

	d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *Scénario : Créer une dimension avec des éléments*.

tPaloCube

Propriétés du tPaloCube

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloCube** crée, supprime ou vide des cubes Palo à partir de dimensions existantes dans une base de données Palo.	
Objectif	Ce composant effectue des opérations sur un cube Palo donné.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données dans laquelle l'opération donnée doit s'effectuer.
	Cube	Saisissez le nom du cube dans lequel l'opération donnée doit s'effectuer.
	Cube type	Sélectionnez le type de cube dans la liste déroulante pour le cube sur lequel l'opération donnée doit s'effectuer : - **Normal** : Type de cube "normal", type par défaut. - **Attribut** : un cube **Attribute** sera créé avec la création d'un cube "normal". - **User Info** : un cube **User Info** sera créé/modifié par ce composant.
	Action on cube	Sélectionnez l'opération que vous souhaitez effectuer sur le cube défini : - **Create cube** : le cube n'existe pas et sera créé. - **Create cube if not exists** : crée un cube s'il n'existe pas. - **Delete cube if exists and create** : supprime un cube s'il existe et en crée un nouveau. - **Delete cube** : supprime le cube de la base de données. - **Clear cube** : supprime les données du cube.
	Dimension list	Ajoutez des lignes en cliquant sur le bouton **[+]** et dans chaque ligne, saisissez le nom d'une dimension existant dans la base de

		données et devant être utilisée dans le cube. L'ordre de cette liste définit également l'ordre de création des dimensions.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en standalone dans la création dynamique de cubes, avec une liste définie de dimensions.	
Connections	Liens de sortie (de ce composant à un autre) : **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Pour plus d'informations concernant les connexions, consultez la section relative aux différents types de connexions, dans le *Guide utilisateur du Studio Talend*.	
Global Variables	**CUBENAME** : nom du cube. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Le cube créant un processus ne crée pas les dimensions à partir de rien. Les dimensions utilisées dans le cube ont été créées précédemment. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Créer un cube dans une base de données existante

Le Job de ce scénario crée un nouveau cube à deux dimensions dans la base de données démo *Biker* de Palo.

Pour reproduire ce scénario, procédez comme suit :

Configurer le composant tPaloCube

1. Déposez un composant **tPaloCube** de la **Palette** dans l'espace de modélisation graphique.
2. Double-cliquez sur le composant **tPaloCube** afin d'ouvrir sa vue **Component**.

3. Dans le champ **Host name**, saisissez le nom de l'hôte ou l'adresse IP du serveur hôte, *localhost* dans cet exemple.
4. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.
5. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.
6. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle vous souhaitez créer le cube, *Biker* dans cet exemple.

7. Dans le champ **Cube**, saisissez le nom que vous souhaitez utiliser afin de créer le cube, *bikerTalend*, par exemple.

8. Dans le champ **Cube type**, sélectionnez le type **Normal** dans la liste déroulante pour le cube à créer, ce qui signifie que ce cube sera de type "Normal", le type par défaut.

9. Dans le champ **Action on cube**, sélectionnez l'action à effectuer. Dans ce scénario, sélectionnez **Create cube**.

10. Sous le tableau **Dimensionlist**, cliquez deux fois sur le bouton **[+]** afin d'ajouter deux lignes.

11. Dans le tableau **Dimensionlist**, saisissez le nom de chaque nouvelle ligne afin de remplacer le nom par défaut. Dans ce scénario, saisissez *Months* pour la première ligne et *Products* pour la seconde. Ces deux dimensions existent déjà dans la base de données *Biker* dans laquelle le nouveau cube sera créé.

Exécuter le Job

Appuyez sur **F6** afin d'exécuter le Job.

Un nouveau cube a été créé dans la base de données *Biker* et les deux dimensions sont ajoutées dans ce cube.

tPaloCubeList

Propriétés du tPaloCubeList

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloCubeList** récupère les informations d'une liste de cubes dans la base de données Palo déterminée.	
Objectif	Ce composant liste le nom des cubes, leur types, le nombre de dimensions assignées, le nombre de cellules remplies dans la base de données déterminée.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données contenant les cubes dont vous souhaitez récupérer les informations.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en tant que composant de début. Il requiert un composant de sortie.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate. **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_CUBES** : nombre de cubes. Cette variable est une variable **After** et retourne un nombre entier. **CUBEID** : ID du cube. Cette variable est une variable **Flow** et retourne un nombre entier. **CUBENAME** : nom du cube. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Le schéma de sortie est fixe et en lecture seule. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Découvrir le schéma de sortie en lecture seule du tPaloCubeList

Le tableau ci-dessous présente les informations relatives au schéma en lecture seule du composant **tPaloCubeList**.

Colonne	Type	Description
Cube_id	int	ID interne du cube.
Cube_name	string	Nom du cube.
Cube_dimensions	int	Nombre de dimensions dans le cube.
Cube_cells	long	Nombre de cellules à l'intérieur du cube.
Cube_filled_cells	long	Nombre de cellules remplies dans le cube.
Cube_status	int	Statut du cube. Il peut être : - **0** : non chargé - **1** : chargé - **2** : changé
Cube_type	int	Type du cube. Il peut être : - **0** : normal - **1** : system - **2** : attribute - **3** : user info - **4** : gpu type

Scénario : Récupérer des informations détaillées d'un cube d'une base de données déterminée

Le Job dans ce scénario récupère des informations détaillées concernant les cubes de la base de données démo de Palo *Biker*.

Pour reproduire ce scénario, procédez comme suit :

Configurer le Job

1. Déposez un composant **tPaloCubeList** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Cliquez-droit sur le **tPaloCubeList** afin d'ouvrir son menu contextuel.
3. Dans ce menu, sélectionnez **Row** > **Main** pour relier les deux composants.

Configurer le composant tPaloCube

1. Double-cliquez sur le **tPaloCube** afin d'ouvrir sa vue **Component**.

2. Dans le champ **Host name**, saisissez le nom de l'hôte ou l'adresse IP du serveur hôte, *localhost* dans cet exemple.
3. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.
4. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.
5. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle vous souhaitez créer le cube, *Biker* dans cet exemple.

Exécuter le Job

Appuyez sur **F6** pour exécuter le Job.

Les informations du cube sont récupérées de la base de données *Biker* et sont affichées dans la console de la vue **Run**.

```
[statistics] connected
26|Orders|8|45291236390400|133005|1|0
27|bikerTalend|2|11830|0|1|0
1|#_GROUP_CUBE_DATA|2|8|0|1|1
2|#_CONFIGURATION|1|2|2|1|1
2|#_CONFIGURATION|1|2|2|1|1
2|#_CONFIGURATION|1|2|2|1|1
6|#_SUBSET_LOCAL|3|0|0|1|1
6|#_SUBSET_LOCAL|3|0|0|1|1
7|#_SUBSET_GLOBAL|2|0|0|1|1
7|#_SUBSET_GLOBAL|2|0|0|1|1
8|#_VIEW_LOCAL|3|0|0|1|1
8|#_VIEW_LOCAL|3|0|0|1|1
9|#_VIEW_GLOBAL|2|0|0|1|1
9|#_VIEW_GLOBAL|2|0|0|1|1
11|#_GROUP_DIMENSION_DATA_Products|2|1352|0|1|1
13|#_GROUP_DIMENSION_DATA_Customers|2|1864|0|1|1
```

Pour plus d'informations concernant l'interprétation des informations sur le cube, consultez la *Découvrir le schéma de sortie en lecture seule du tPaloCubeList*.

tPaloDatabase

Propriétés du tPaloDatabase

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloDatabase** crée, supprime ou recrée des bases de données sur un serveur Palo donné.	
Objectif	Ce composant gère les bases de données dans un serveur Palo.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données dans laquelle l'opération doit s'effectuer.
	Action on database	Sélectionnez l'opération que vous souhaitez effectuer sur le base de données: - **Create database** : la base de données n'existe pas et sera créée. - **Create database if not exists** : la base est créée lorsqu'elle n'existe pas. - **Delete database if exists and create** : la base de données est supprimée si elle existe et une nouvelle base de données est créée. - **Delete database** : la base de données est supprimée du serveur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**DATABASENAME** : nom de la base de données. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Connections	Liens de sortie (de ce composant à un autre) : **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error Pour plus d'informations concernant les connexions, consultez la section relative aux différents types de connexions dans le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone lors d'opérations de gestion de bases de données dans un serveur Palo.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Créer une base de données

Le Job de ce scénario crée une nouvelle base de données sur un serveur Palo donné.

Pour reproduire ce scénario, procédez comme suit :

1. Déposez un composant **tPaloDatabase** de la **Palette** dans l'espace de modélisation graphique.
2. Double-cliquez sur le composant **tPaloDatabase** afin d'ouvrir sa vue **Component**.

3. Dans le champ **Host name**, saisissez le nom de l'hôte ou l'adresse IP du serveur hôte, *localhost* dans cet exemple.

4. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.

5. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.

6. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle vous souhaitez créer le cube, *talenddatabase* dans cet exemple.

7. Dans le champ **Action on database**, sélectionnez l'action à effectuer. Dans ce scénario, sélectionnez **Create database** puisque la base de données à créer n'existe pas.

8. Appuyez sur **F6** pour exécuter votre Job.

Une nouvelle base de données est créée sur le serveur Palo donné.

tPaloDatabaseList

Propriétés du tPaloDatabaseList

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloDatabaseList** récupère les informations d'une liste de bases de données dans un serveur Palo donné.	
Objectif	Ce composant liste les noms des bases de données, leur type, le nombre de cubes, le nombre de dimensions, le statut et l'ID des bases de données d'un serveur Palo donné.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en tant que composant de début. Il nécessite un composant de sortie.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate. **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Pour plus d'informations concernant les connexions, consultez la section relative aux types de connexions dans le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_DATABASES** : nombre de bases de données. Cette variable est une variable **After** et retourne un nombre entier. **DATABASEID** : ID de la base de données. Cette variable est une variable **Flow** et retourne un long. **DATABASENAME** : nom de la base de données. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Le schéma de sortie est fixe et en lecture seule. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Découvrir le schéma de sortie en lecture seule du tPaloDatabaseList

Le tableau ci-dessous présente les informations relatives au schéma de sortie en lecture seule du composant **tPaloDatabaseList**.

Base de données	Type	Description
Database_id	long	ID interne de la base de données.
Database_name	string	Nom de la base de données.
Database_dimensions	int	Nombre de dimensions dans la base de données.
Database_cubes	int	Nombre de cubes dans la base du données.
Database_status	int	Statut de la base de données. - **0** = non chargée - **1** = chargée - **2** = changée
Database_types	int	Type de la base de données. - **0** = normal - **1** = system - **3** = user info

Scénario : Récupérer des informations détaillées concernant les bases de données d'un serveur Palo donné

Le Job de ce scénario récupère des informations concernant toutes les bases de données d'un serveur Palo donné.

Pour reproduire ce scénario, procédez comme suit :

Configurer le Job

1. Déposez un composant **tPaloDatabaseList** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Cliquez-droit sur le **tPaloDatabaseList** pour ouvrir le menu contextuel.
3. Dans ce menu, sélectionnez **Row** > **Main** pour relier les deux composants.

Configurer le composant tPaloDatabaseList

1. Double-cliquez sur le **tPaloDatabaseList** pour ouvrir sa vue **Component**.

2. Dans le champ **Host name**, saisissez le nom de l'hôte ou l'adresse IP du serveur hôte, *localhost* dans cet exemple.
3. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.
4. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.

Exécuter le Job

Appuyez sur **F6** pour exécuter le Job.

Les informations de toutes les bases de données du serveur Palo sont récupérées et affichées dans la console de la vue **Run**.

```
0|System|7|6|1|1
1|Demo|23|23|1|0
6|Biker|27|28|1|0
9|tx_g|17|17|1|0
25|DBHSE24|13|12|1|0
31|elmCreate|13|12|1|0
32|tPaloOutputMulit|15|15|1|0
```

Pour plus d'informations concernant le schéma de sortie, consultez *Découvrir le schéma de sortie en lecture seule du tPaloDatabaseList*.

tPaloDimension

Propriétés du tPaloDimension

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloDimension** crée, supprime ou recrée des dimensions avec ou sans éléments de dimension dans une base de données Palo.	
Objectif	Ce composant gère les dimensions Palo, ainsi que les éléments dans une base de données.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données dans laquelle les dimensions sont gérées.
	Dimension	Saisissez le nom de la dimension sur laquelle l'opération donnée doit s'effectuer.
	Action on dimension	Sélectionnez l'opération que vous souhaitez effecteur sur la dimension : - **None** : aucune action n'est effectuée sur la dimension. - **Create dimension** : la dimension n'existe pas et sera créée. - **Create dimension if not exists** : cette dimension est créée quand elle n'existe pas. - **Delete dimension if exists and create** : cette dimension est supprimée si elle existe et une nouvelle dimension est créée. - **Delete dimension** : la dimension est supprimée de la base de données.
	Create dimension elements	Cochez cette case pour activer les champs de gestion des dimensions et créer des éléments de dimension , et créer cette dimension.
Les champs ci-dessous sont disponibles uniquement quand la case **Create dimension elements** est cochée.	*Dimension type* Disponible uniquement lorsque l'action	Sélectionnez le type de dimension à créer. Le type peut être : - **Normal** - **User info** - **System**

	à effectuer sur la dimension est **None**.	- **Attribute**
	Commit size	Saisissez le nombre d'éléments qui seront créés avant de les sauvegarder dans la dimension.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Consolidation type - None Cette option active les champs des paramètres correspondants devant être renseignés.	Sélectionnez ce bouton pour déplacer directement les éléments entrants dans la dimension donnée. Avec cette option, vous ne définissez aucune consolidation ou hiérarchie.
		Input Column : sélectionnez une colonne dans la liste déroulante. Les colonnes de la liste déroulante sont celles que vous avez définies dans le schéma. Les valeurs de la colonne sélectionnée seront prises en compte lors du traitement d'éléments de dimension.
		Element type : Sélectionnez le type d'éléments. Il peut être : - **Numeric** - **Text**
		Creation mode : Sélectionnez le mode de création pour les éléments à traiter. Ce mode peut être : - **Add** : ajoute simplement un élément à la dimension. - **Force add** : force la création de cet élément. S'il existe, l'élément sera recréé. - **Update** : met à jour l'élément s'il existe. - **Add or Update** : si l'élément n'existe pas, il sera créé. S'il existe, il sera mis à jour. Cette option est l'option par défaut. - **Delete** : supprime cet élément de la dimension.

	Consolidation type - Normal Cette option active les champs des paramètres correspondants devant être renseignés.	Sélectionnez ce bouton pour créer des éléments et les consolider dans la dimension donnée. Cette consolidation structure les éléments créés à différents niveaux.
		Input Column: sélectionnez une colonne dans la liste déroulante. Les colonnes de la liste déroulante sont celles que vous avez définies dans le schéma. Les valeurs de la colonne sélectionnée seront prises en compte lors du traitement d'éléments de dimension.
		Element type : Sélectionnez le type d'éléments. Il peut être : - **Numeric** - **Text**
		Creation mode : Sélectionnez le mode de création pour les éléments à créer. Ce mode peut être : - **Add** : ajoute simplement un élément à la dimension. - **Force add** : force la création de cet élément. S'il existe, l'élément sera recréé. - **Update** : met à jour l'élément s'il existe. - **Add or Update** : si l'élément n'existe pas, il sera créé. S'il existe, il sera mis à jour. Cette option est l'option par défaut.
	Consolidation type - Self-referenced Cette option active les champs des paramètres correspondants devant être renseignés.	Sélectionnez ce bouton pour créer des éléments et les structurer à partir d'une relation parent-enfant. Le flux d'entrée est responsable du regroupement de la consolidation.
	Element's type	Sélectionnez le type d'éléments. Il peut être : - **Numeric** - **Text**
	Creation mode	Sélectionnez le mode de création pour les éléments à créer. Ce mode peut être : - **Add** : ajoute simplement un élément à la dimension. - **Force add** : force la création de cet élément. S'il existe, l'élément sera recréé. - **Update** : met à jour l'élément s'il existe. - **Add or Update** : si l'élément n'existe pas, il sera créé. S'il existe, il sera mis à jour. Cette option est l'option par défaut.
		Input Column : sélectionnez une colonne dans la liste déroulante. Les colonnes dans cette liste sont celles que vous avez définies dans le schéma. Les valeurs de la colonne sélectionnée seront prises en compte lors du traitement d'éléments de dimension.
		Hierarchy Element : sélectionnez le type et la relation de cette colonne d'entrée pour la consolidation. - **Parent** : définissez la valeur d'entrée en tant qu'élément parent. - **Child** : lie la valeur d'entrée à la valeur parent et construit la consolidation.

		- **Factor** : définissez le facteur de cette consolidation.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en standalone ou comme composant de fin d'un processus.	
Connections	Liens de sortie (de ce composant à un autre) : **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Iterate **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error Pour plus d'informations concernant les connexions, consultez la section relatives aux différents types de connexions, dans le *Guide utilisateur du Studio Talend*.	
Global Variables	**DIMENSIONNAME** : nom de la dimension. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	La suppression d'éléments de dimension est possible uniquement avec le type de consolidation **None**. Seul le type de consolidation **Self-Referenced** permet le placement d'un facteur dans cette consolidation. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Créer une dimension avec des éléments

Le Job de ce scénario crée une dimension de date avec une hiérarchie d'éléments simple, composée de trois niveaux : *Year, Month, Date* (Année, mois, date).

Pour reproduire ce scénario, procédez comme suit :

Construire le Job

1. Déposez un composant **tPaloConnection**, un **tRowGenerator**, un **tMap**, et un **tPaloDimension** de la **Palette** dans l'espace de modélisation graphique.

2. Cliquez-droit sur le **tPaloConnection** afin d'ouvrir le menu contextuel et sélectionnez **Trigger** > **On Subjob Ok** pour le relier au **tRowGenerator**.

3. Cliquez-droit sur le **tRowGenerator** afin d'ouvrir le menu contextuel et sélectionnez **Row** > **Main** pour le relier au **tMap**.

   Le **tRowGenerator** est utilisé pour générer des lignes au hasard afin de simplifier ce processus. Dans un cas réel d'utilisation, vous pouvez utiliser un autre composant d'entrée afin de charger vos données.

4. Cliquez-droit sur le **tMap** pour ouvrir le menu contextuel et sélectionnez **Row** > ***New output*** pour le relier au **tPaloDimension**. Nommez ce lien *out1* dans la boîte de dialogue qui s'ouvre.

Paramétrer la connexion à une base de données

1. Double-cliquez sur le composant **tPaloCube** afin d'ouvrir sa vue **Component**.

2. Dans le champ **Host name**, saisissez le nom de l'hôte ou l'adresse IP du serveur hôte, *localhost* dans cet exemple.

3. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.

4. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez les informations d'authentification. Dans cet exemple, les deux sont *admin*.

Configurer le composant Input

1. Double-cliquez sur le **tRowGenerator** pour ouvrir son éditeur.

2. Dans la partie supérieure de l'éditeur, cliquez sur le bouton **[+]** pour ajouter une colonne, et renommez-la *random_date* dans la colonne **Column**.

3. Dans la ligne nouvellement ajoutée, sélectionnez **Date** pour la colonne **Type**, et **getRandomDate** dans la colonne **Functions**.

4. Dans la vue **Function parameters** de la partie inférieure de l'éditeur, saisissez les nouvelles valeurs des dates minimum et maximum de la colonne **Value**. Dans cet exemple, le minimum est *2010-01-01*, le maximum est *2010-12-31*.

5. Cliquez sur **OK** afin de valider vos modifications et fermer l'éditeur.

6. Dans la boîte de dialogue qui s'ouvre, cliquez sur **OK** pour propager les modifications.

Configurer le tMap editor

1. Double-cliquez sur le **tMap** afin d'ouvrir son éditeur.

2. Dans la vue **Schema editor** de la partie inférieure de la fenêtre, sous la table *out1*, cliquez sur le bouton **[+]** afin d'ajouter trois lignes.

3. Dans la colonne **Column** de la table *out1* saisissez de nouveaux noms pour les trois lignes ajoutées, *Year*, *Month* et *Date* (année, mois et date). Ces lignes sont ajoutées automatiquement à la table *out1* dans la partie inférieure du **tMap editor**.

4. Dans la table *out1* de la partie supérieure de l'éditeur, cliquez sur la colonne **Expression** de la ligne *Year*.

5. Appuyez sur les touches **Ctrl+Espace** afin d'ouvrir la liste d'auto-complétion.

6. Double-cliquez sur TalendDate.formatDate pour la sélectionner. L'expression pour obtenir la date s'affiche dans la ligne Year. L'expression est la suivante :

   `TalendDate.formatDate("yyyy-MM-dd HH:mm:ss",myDate).`

7. Remplacez l'expression par défaut par `TalendDate.formatDate("yyyy",row1.random_date).`

8. Répétez l'opération pour les lignes Month et Date pour ajouter l'expression par défaut et remplacez-la respectivement par `TalendDate.formatDate("MM",row1.random_date)`et par `TalendDate.formatDate("dd-MM-yyyy", row1.random_date).`

9. Cliquez sur **OK** afin de valider cette modification et acceptez la propagation des modifications en cliquant sur **OK** dans la boîte de dialogue qui s'ouvre.

Configurer le composant tPaloDimension

1. Dans l'espace de modélisation graphique, double-cliquez sur le **tPaloDimension** afin d'ouvrir sa vue **Component**.

2. Cochez la case **Use an existing connection**. Le **tPaloConnection_1** s'affiche automatiquement dans le champ **Connection configuration**.

3. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle la nouvelle dimension est créée, *talendDatabase* dans cet exemple.

4. Dans le champ **Dimension**, saisissez le nom que vous souhaitez utiliser pour la dimension à créer, *Date*, par exemple.

5. Dans le champ **Action on dimension**, sélectionnez l'action à effectuer. Dans ce scénario, sélectionnez **Create dimension if not exist**.

6. Sélectionnez la case **Create dimension elements**.

7. Dans la zone **Consolidation Type**, sélectionnez le bouton **Normal**.

8. Sous le tableau de la hiérarchie des éléments, cliquez trois fois sur le bouton **[+]** pour ajouter trois lignes au tableau.

9. Dans la colonne **Input column** du tableau, sélectionnez *Year* dans la liste déroulante de la première ligne, *Month* dans la deuxième et *Date* dans la troisième. Cela détermine le niveau des éléments des différentes colonnes du schéma d'entrée.

Exécuter le Job

Appuyez sur **F6** pour exécuter le Job.

Une nouvelle dimension a été créée dans la base de données Palo *talendDatabase*.

Modeller - LOCALHOST / talendDatabase
Database
Date
2010
04
26-04-2010
25-04-2010
30-04-2010
09-04-2010
15-04-2010
04-04-2010
02-04-2010
12-04-2010
23-04-2010
11
01
19-01-2010
17-01-2010
15-01-2010
29-01-2010
08-01-2010
12-01-2010
06-01-2010

tPaloDimensionList

Propriétés du tPaloDimensionList

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloDimensionList** récupère les informations d'une liste de dimensions de la base de données Palo déterminée.	
Objectif	Ce composant affiche le nom des dimensions, leur type, le nombre d'éléments de dimension, le maximum d'indentation des dimensions, le maximum de profondeur des dimensions, le maximum des niveaux des dimensions, l'ID des dimensions dans un serveur Palo donné.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données dans laquelle se trouvent les dimensions qui vous intéressent.
	Retrieve cube dimensions	Cochez cette case pour récupérer les informations des dimensions d'un cube existant.
	Cube Disponible lorsque vous cochez la case **Retrieve cube dimensions**.	Saisissez le nom du cube duquel les informations concernant les dimensions sont récupérées.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en standalone ou comme composant de début d'un processus.	
Connections		Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate. **Trigger** : Run if, On Subjob Ok; On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate. **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Pour plus d'informations concernant les connexions, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.
Global Variables	**DIMENSIONNAME** : nom de la dimension. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Le schéma de sortie est fixe et en lecture seule. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Découvrir le schéma de sortie en lecture seule du tPaloDimensionList

Le tableau ci-dessous présente les informations relatives au schéma de sortie en lecture seule du composant **tPaloDimensionList**.

Dimension	Type	Description
Dimension_id	long	ID interne de la dimension.
Dimension_name	string	Nom de la dimension.
Dimension_attribute_cube	string	Nom du cube des attributs.
Dimension_rights_cube	string	Nom du cube des droits.
Dimension_elements	int	Nombre d'éléments de dimension.
Dimension_max_level	int	Niveau maximal de la dimension.
Dimension_max_indent	int	Indentation maximum de la dimension.
Dimension_max_depth	int	Profondeur maximale de la dimension.
Dimension_type	int	Type de la dimension : - **0** = normal - **1** = system - **2** = attribute - **3** = user info

Scénario: Récupérer des informations détaillées concernant les dimensions d'une base de données déterminée

Le Job de ce scénario récupère les informations de toutes les dimensions d'une base de données déterminée.

Pour reproduire ce scénario, procédez comme suit :

Construire le Job

1. Déposez un composant **tPaloDimensionList** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Cliquez-droit sur le **tPaloDimensionList** pour ouvrir le menu contextuel.
3. Dans le menu, sélectionnez **Row** > **Main** pour relier les deux composants.

Configurer le composent tPaloDimensionList

1. Double-cliquez sur le composant **tPaloDimensionList** pour ouvrir sa vue **Component**.

2. Dans le champ **Host name**, saisissez le nom de l'hôte ou l'adresse ID du serveur hôte, *localhost* dans cet exemple.

3. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.

4. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.

5. Dans le champ **Database**, saisissez le nom de la base de données contenant les dimensions qui vous intéressent, *Biker* dans cet exemple.

Exécuter le Job

Appuyez sur **F6** pour exécuter le Job.

Les informations concernant les dimensions de la base de données *Biker* sont récupérées et affichées dans la console de la vue **Run**.

```
[statistics] connected
11|Products||#_Products|#_GROUP_DIMENSION_DATA_Products|338|3|4|3|0
13|Customers||#_Customers|#_GROUP_DIMENSION_DATA_Customers|466|3|4|3|0
15|Orderlines||#_Orderlines|#_GROUP_DIMENSION_DATA_Orderlines|3890|1|2|1|0
17|Channels||#_Channels|#_GROUP_DIMENSION_DATA_Channels|4|1|2|1|0
19|Years||#_Years|#_GROUP_DIMENSION_DATA_Years|11|1|2|1|0
21|Months||#_Months|#_GROUP_DIMENSION_DATA_Months|35|18|18|18|0
23|Datatypes||#_Datatypes|#_GROUP_DIMENSION_DATA_Datatypes|8|1|2|1|0
25|Measures||#_Measures|#_GROUP_DIMENSION_DATA_Measures|6|1|2|1|0
0|#_USER_||||5|0|1|0|0
1|#_GROUP_||||4|0|1|0|0
2|#_CUBE_||||2|0|1|0|0
4|#_CONFIGURATION_||||2|0|1|0|0
5|#_DIMENSION_||||8|0|1|0|0
7|#_SUBSET_||#_#_SUBSET_||0|0|0|0|0
9|#_VIEW_||#_#_VIEW_||0|0|0|0|0
3|#_#_CUBE__|||0|0|0|0|0
6|#_#_DIMENSION__||||0|0|0|0|0
8|#_#_SUBSET__||||0|0|0|0|0
10|#_#_VIEW__||||0|0|0|0|0
12|#_Products_||||0|0|0|0|0
14|#_Customers_||||0|0|0|0|0
16|#_Orderlines_||||0|0|0|0|0
18|#_Channels_||||0|0|0|0|0
20|#_Years_||||0|0|0|0|0
```

Pour plus d'informations concernant le schéma de sortie, consultez *Découvrir le schéma de sortie en lecture seule du tPaloDimensionList*.

tPaloInputMulti

Propriétés du tPaloInputMulti

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloInputMulti** récupère des données (des éléments et des valeurs) d'un cube Palo.	
Objectif	Ce composant récupère les valeurs stockées ou calculées ainsi que les enregistrements à l'extérieur d'un cube.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données dans laquelle se trouvent les éléments qui vous intéressent.
	Cube	Saisissez le nom du cube dans lequel les éléments de dimension à récupérer sont stockés.
	Cube type	Sélectionnez le type de cube dans la liste déroulante pour le cube sur lequel l'opération donnée doit s'effectuer : - **Normal** : Type de cube "normal", type par défaut. - **Attribut** : un cube **Attribute** sera créé avec la création d'un cube "normal". - **System** : Type de cube "système". - **User Info** : un cube **User Info** sera créé/modifié par ce composant.
	Commit size	Saisissez le nombre de lignes de chaque lot à récupérer.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Cube Query	Remplissez ce tableau pour préciser les données que vous souhaitez récupérer.
		Column : les colonnes du schéma sont ajoutées automatiquement à cette colonne, une fois définies dans l'éditeur du schéma. Les colonnes du schéma sont utilisées pour stocker les éléments de dimensions récupérés.
		Dimensions : saisissez le nom de chaque dimension du cube duquel vous souhaitez récupérer les éléments de dimensions. *L'ordre des dimensions dans cette colonne doit être cohérent avec l'ordre dans le cube stockant ces dimensions.*
		Elements : saisissez les éléments de dimensions desquels récupérer les données. Si plusieurs éléments d'une dimension sont requis, séparez-les avec une virgule.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant requiert un composant de sortie.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate. **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Selon l'architecture des systèmes OLAP, une seule valeur (texte ou numérique) peut être récupérée du cube. Les colonnes **MEASURE** et **TEXT** sont fixes et en lecture seule. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Récupérer des éléments de dimensions dans un cube donné

Le Job de ce scénario récupère plusieurs éléments de dimensions du cube de démo Palo *Sales*.

Pour reproduire ce scénario, procédez comme suit :

Construire le Job

1. Déposez un composant **tPaloInputMulti** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Cliquez-droit sur le composant **tPaloInputMulti** afin d'ouvrir son menu contextuel.
3. Dans le menu, sélectionnez **Row** > **Main** afin de relier le **tPaloInputMulti** au **tLogRow**.

Paramétrer la connexion à une base de données

1. Double-cliquez sur le **tPaloInputMulti** afin d'ouvrir sa vue **Component**.

2. Dans le champ **Host name**, saisissez le nom de l'hôte ou l'adresse IP du serveur hôte, *localhost* dans cet exemple.

3. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.

4. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, l'identifiant et le mot de passe sont *admin*.

Configurer le Cube Query

1. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle est stocké le cube.

2. Dans le champ **Cube**, saisissez le nom du cube dans lequel sont stockées les dimensions qui vous intéressent Dans ce scénario, le cube est celui de démo, *Sales*.

3. Dans le champ **Cube type**, sélectionnez le type du cube à créer. Choisissez **Normal**, ce qui signifie que le cube sera de type "Normal", le type par défaut.

4. A côté du champ **Edit Schema**, cliquez sur le bouton **[...]** pour ouvrir l'éditeur de schéma.

5. Dans l'éditeur du schéma, cliquez sur le bouton **[+]** pour ajouter les lignes nécessaires. Dans cet exemple, ajoutez les lignes correspondant à toutes les dimensions stockées dans le cube *Sales* : *Products*, *Regions*, *Months*, *Years*, *Datatypes*, *Measures*. Saisissez les noms dans l'ordre donné de ce cube.

6. Cliquez sur **OK** pour valider les modifications et acceptez la propagation du changement au composant suivant. Ces colonnes sont ajoutées automatiquement dans la colonne **Column** du tableau **Cube Query** dans la vue **Component**. Si l'ordre n'est pas cohérent par rapport à celui du cube *Sales*, déplacez les colonnes à l'aide des flèches montante et descendante.

7. Dans la colonne **Dimensions** du tableau **Cube Query**, saisissez le nom de chaque dimension stockée dans le cube *Sales* selon l'ordre des lignes dans la colonne **Column**. Dans le cube *Sales*, le nom des dimensions est : *Products*, *Regions*, *Months*, *Years*, *Datatypes*, *Measures*.

8. Dans la colonne **Elements** du tableau **Cube Query**, saisissez le nom des éléments de dimensions que vous souhaitez récupérer selon la dimension à laquelle ils appartiennent. Dans cet exemple, les éléments à récupérer sont *All products*, *Germany*, *Austria*, *Jan*, *2009*, *Actual*, *Turnover* (*Germany* et *Austria* appartenant à la même dimension **Regions**, ils doivent être saisis dans la même ligne et séparés par une virgule).

Exécuter le Job

1. Cliquez sur le **tLogRow** afin d'ouvrir sa vue **Component**.

2. Dans la zone **Mode**, sélectionnez l'option **Table (print values in cell of a table)** pour afficher les résultats de l'exécution sous forme de tableau.

3. Appuyez sur **F6** pour exécuter le Job.

Les éléments de dimensions et la valeur **Measure** correspondant s'affichent dans la console de la vue **Run**.

```
Starting job retrieve at 13:49 02/11/2010.

[statistics] connecting to socket on port 3338
[statistics] connected
.------------+-------+------+-----+---------+--------+---------+----.
|                                tLogRow_1                            |
|=-----------+-------+------+-----+---------+--------+---------+---=|
|Products    |Regions|Months|Years|Datatypes|Measures|MEASURE  |TEXT|
|=-----------+-------+------+-----+---------+--------+---------+---=|
|All Products|Austria|Jan   |2009 |Actual   |Turnover|476977.95|null|
|All Products|Germany|Jan   |2009 |Actual   |Turnover|1234.56  |null|
'------------+-------+------+-----+---------+--------+---------+----'
[statistics] disconnected
Job retrieve ended at 13:49 02/11/2010. [exit code=0]
```

tPaloOutput

Propriétés du tPaloOutput

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloOutput** écrit une ligne de données (des éléments ainsi que des valeurs) dans un cube Palo donné.	
Objectif	Ce composant prend le flux d'entrée et l'écrit dans un Cube Palo donné.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données dans laquelle se trouve le cube qui vous intéresse.
	Cube	Saisissez le nom du cube dans lequel les données entrantes sont écrites.
	Commit size	Saisissez le nombre de lignes de chaque lot devant être écrit dans le cube.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Column as Measure	Sélectionnez la colonne du flux d'entrée contenant les valeurs **Measure** ou **Text**.
	Create element if not exist	Cochez cette case afin de créer l'élément à traiter s'il n'existe pas.
	Save cube at process end	Cochez cette case afin de sauvegarder le cube dans lequel vous avez écrit les données à la fin du processus.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant requiert un composant d'entrée.	
Global variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Iterate. **Trigger** : Run if. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant ne peut écrire qu'une seule ligne de données dans un cube. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *Scénario 1 : Ecrire des données dans un cube donné*.

tPaloOutputMulti

Propriétés du tPaloOutputMulti

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloOutputMulti** écrit des données (des éléments et des valeurs) dans un cube Palo.	
Objectif	Ce composant prend le flux d'entrée et l'écrit dans un cube Palo donné.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données dans laquelle se trouve le cube qui vous intéresse.
	Cube	Saisissez le nom du cube dans lequel les données entrantes sont écrites.
	Cube type	Sélectionnez le type de cube dans la liste déroulante pour le cube sur lequel l'opération donnée doit s'effectuer : - **Normal** : Type de cube "normal", type par défaut. - **Attribut** : un cube **Attribute** sera créé avec la création d'un cube "normal". - **System** : Type de cube "système". - **User Info** : un cube **User Info** sera créé/modifié par ce composant.
	Commit size	Saisissez le nombre de lignes de chaque lot devant être écrit dans le cube.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Measure value	Sélectionnez la colonne du flux d'entrée contenant les valeurs **Measure** ou **Text**.
	Splash mode	Sélectionnez le mode splash utilisé pour écrire des données dans un élément consolidé. Le mode peut être : - **Add** :écrit des valeurs dans les éléments sous-jacents. - **Default** : utilise le mode splash par défaut. - **Set** : définit simplement la valeur et effectue la distribution à partir d'autres valeurs. - **Disable** : n'applique aucun splash. Pour plus d'informations concernant les modes splash Palo, consultez le Guide utilisateur de Palo.
	Add values	Cochez cette case pour ajouter des nouvelles valeurs aux valeurs actuelles. Si vous ne cochez pas cette case, les nouvelles valeurs écraseront les valeurs actuelles.
	Use eventprocessor	Cochez cette case pour appeler le serveur de supervision.
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant requiert un composant de début.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main **Trigger** : Run if, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Les mesures numériques doivent être de type Double ou String. Lorsque le type String est utilisé, saisissez entre guillemets la valeur devant être traitée. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario 1 : Ecrire des données dans un cube donné

Le Job de ce scénario écrit de nouvelles valeurs dans le cube *Sales* donné en démo dans la base de données *Demo* installée avec Palo.

Pour reproduire ce scénario, procédez comme suit :

Construire le Job

1. Déposez un **tFixedFlowInput** et un **tPaloOutputMulti** de la **Palette** dans l'espace de modélisation graphique.
2. Cliquez-droit sur le composant **tFixedFlowInput** afin d'ouvrir son menu contextuel.
3. Dans ce menu, sélectionnez **Row** > **Main** pour relier ce composant au **tPaloOutputMulti**.

Configurer le composant d'entrée

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Component**.

2. Cliquez sur le bouton **[...]** pour ouvrir l'éditeur de schéma.

3. Dans l'éditeur de schéma, cliquez sept fois sur le bouton **[+]** afin d'ajouter sept lignes. Renommez-les respectivement *Products*, *Regions*, *Months*, *Years*, *Datatypes*, *Measures* et *Values*. L'ordre de ces lignes doit être cohérent avec celui des dimensions correspondantes dans le cube *Sales* et le type de la colonne *Value*, qui comprend la valeur de la mesure, est défini comme double/Double.

4. Cliquez sur **OK** afin de valider la modification et acceptez la propagation proposée par la boîte de dialogue qui s'ouvre. Les libellés des colonnes du schéma s'affichent automatiquement dans le tableau **Value** sous la case **Use single table** de la zone **Mode**.

5. Dans le tableau **Value**, saisissez la valeur de chaque ligne dans la colonne **Value**. Ces valeurs sont : *Desktop L*, *Germany*, *Jan*, *2009*, *Actual*, *turnover*, *1234.56*.

Configurer le composant de sortie

1. Double-cliquez sur le composant **tPaloOutputMulti** afin d'ouvrir sa vue **Component**.

2. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.

3. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.

4. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle vous souhaitez créer le cube, *Biker* dans cet exemple.

5. Dans le champ **Cube**, saisissez le nom que vous souhaitez utiliser pour le cube à créer, *bikerTalend*, par exemple.

6. Dans le champ **Cube type**, sélectionnez le type **Normal** dans la liste déroulante. Cela signifie que le cube à créer sera de type "Normal", le type par défaut.

7. Dans le champ **Measure Value**, sélectionnez **Value**.

Exécuter le Job

Appuyez sur **F6** pour exécuter le Job.

Le flux d'entrée de données a été écrit dans le cube *Sales*.

	A
1	localhost/Demo
2	Sales
3	Desktop L
4	**Germany**
5	Jan
6	2009
7	Actual
8	Turnover
9	
10	1 234,56
11	

Scénario 2 : Refuser un flux de données d'entrée lorsque les éléments à écrire n'existent pas dans un cube donné

Le Job de ce scénario essaye d'écrire des données dans le cube *Sales* mais puisque les éléments concernés n'existent pas dans ce cube, le flux d'entrée est refusé.

Pour reproduire ce scénario, procédez comme suit :

Construire le Job

1. Déposez les composants **tFixedFlowInput**, **tPaloCheckElements**, **tPaloOutputMulti** et **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Cliquez-droit sur le **tFixedFlowInput** afin d'ouvrir son menu contextuel.
3. Dans ce menu, sélectionnez **Row** > **Main** afin de connecter ce composant au **tPaloCheckElements**.

Configurer le composant d'entrée

1. Double-cliquez sur le composant **tFixedFlowInput** afin d'ouvrir sa vue **Component**.

2. Cliquez sur le bouton **[...]** afin d'ouvrir l'éditeur de schéma.

3. Dans l'éditeur de schéma, cliquez sept fois sur le bouton **[+]** afin d'ajouter sept lignes que vous allez nommer respectivement *Products*, *Regions*, *Months*, *Years*, *Datatypes*, *Measures* et *Values*. L'ordre de ces lignes doit être cohérent avec celui des dimensions correspondantes dans le cube *Sales* et le type de la colonne *Value*, qui comprend la valeur de la mesure et qui est défini comme double/Double.

4. Cliquez sur **OK** afin de valider la modification et acceptez la propagation proposée par la boîte de dialogue qui s'ouvre. Les libellés des colonnes du schéma s'affichent automatiquement dans le tableau **Value** sous la case **Use single table** de la zone **Mode**.

5. Dans le tableau **Value**, saisissez la valeur de chaque ligne dans la colonne **Value**. Ces valeurs sont : *Desktop L*, *Germany*, *Jan*, *2009*, *Actual*, *turnover*, *1234.56*.

Configurer le composant tPaloCheckElements

1. Double-cliquez sur le composant **tPaloCheckElements** afin d'ouvrir sa vue **Component**.

2. Dans le champ **Host name**, saisissez *localhost*.

3. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.

4. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.

5. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle vous souhaitez créer le cube, *Demo* dans cet exemple.

6. Dans le champ **Cube**, saisissez le nom du cube dans lequel vous souhaitez écrire les données, *Sales*, par exemple.

7. Dans le champ **On Element Error**, sélectionnez **Reject** dans la liste déroulante.

8. Dans le tableau contenant les éléments en bas de la vue **Basic settings**, cliquez sur la cellule au croisement de la ligne **Value** et de la colonne **Element type**. Sélectionnez **Measure** dans la liste déroulante.

Configurer le composant de sortie

1. Double-cliquez sur le **tPaloOutputMulti** pour afficher sa vue **Component**.

2. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.

3. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.

4. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle vous souhaitez créer le cube, *Demo* dans cet exemple.

5. Dans le champ **Cube**, saisissez le nom du cube dans lequel vous souhaitez écrire les données, *Sales*, par exemple.

6. Dans le champ **Cube type**, sélectionnez le type **Normal** dans la liste déroulante. Cela signifie que le cube à créer sera de type "Normal", le type par défaut.

7. Dans le champ **Measure Value**, sélectionnez **Value**.

Exécuter le Job

Appuyez sur **F6** pour exécuter votre Job.

Les données à écrire sont rejetées et affichées dans la console de la vue **Run**. Vous pouvez voir que le message d'erreur est *Smart Products*.

```
Starting job JobOutputExistingElements at 14:45 09/11/2010.

[statistics] connecting to socket on port 3407
[statistics] connected
.--------------+-------+------+-----+---------+--------+-------+----------------------
|                                    tLogRow_1
|=-------------+-------+------+-----+---------+--------+-------+----------------------
|Products      |Regions|Months|Years|Datatypes|Measures|Value  |errorMessage
|=-------------+-------+------+-----+---------+--------+-------+----------------------
|Smart Products|Germany|Jan   |2009 |Actual   |Turnover|1234.56|Products:Smart Products
'--------------+-------+------+-----+---------+--------+-------+----------------------
[statistics] disconnected
Job JobOutputExistingElements ended at 14:45 09/11/2010. [exit code=0]
```

tPaloRule

Propriétés du tPaloRule

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloRule** crée ou modifie des règles dans un cube Palo donné.	
Objectif	Ce composant vous permet de gérer des règles dans un cube Palo donné.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données dans laquelle se trouvent les dimensions appliquant les règles.
	Cube	Saisissez le nom du cube dans lequel les informations de dimensions doivent être retrouvées.
	Cube rules	Remplissez ce tableau afin d'effectuer différentes actions sur les règles spécifiques : - **Definition** : saisissez la règle à appliquer.
		- **External Id** : saisissez l'ID externe personnalisé.
		- **Comment** : saisissez un commentaire pour cette règle.
		- **Activated** : cochez cette case afin d'activer la règle.
		- **Action** : sélectionnez dans la liste déroulante l'action à effectuer : - **Create** : créer cette règle. - **Delete** : supprimer cette règle. - **Update** : mettre à jour cette règle.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en standalone dans la création dynamique de cubes, avec une liste définie de dimensions.	
Connections	Liens de sortie (de ce composant à un autre) : **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) :	

	Row : Iterate **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	La mise à jour et la suppression d'une règle ne sont disponibles uniquement lorsque la règle a été créée avec un ID externe. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Créer une règle dans un cube donné

Le Job de ce scénario crée une règle appliquée aux dimensions d'un cube donné.

Pour reproduire ce scénario, procédez comme suit :

Paramétrer la connexion à une base de données

1. Déposez un composant **tPaloRule** de la **Palette** dans l'espace de modélisation graphique.

2. Double-cliquez sur le composant **tPaloRule** afin d'ouvrir sa vue **Component**.

3. Dans le champ **Host name**, saisissez le nom de l'hôte ou l'adresse IP du serveur hôte, *localhost* dans cet exemple.

4. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.

5. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.

6. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle se trouvent les dimensions appliquant les règles, *Biker* dans cet exemple.

7. Dans le champ **Cube**, saisissez le nom du cube auquel appartiennent les dimensions appliquant les règles, *Orders* dans cet exemple.

Définir les règles du cube

1. Sous le tableau **Cube rules**, cliquez sur le bouton **[+]** afin d'ajouter une ligne.

2. Dans le tableau **Cube rules**, saisissez `['2009'] = 123` dans la colonne**Definition**, `OrderRule1`dans la colonne **External Id** et `Palo Demo Rules` dans la colonne **Comment**.

3. Dans la colonne **Activated**, cochez la case.

4. Dans la colonne **Action**, sélectionnez **Create** dans la liste déroulante.

Exécuter le Job

Appuyez sur **F6** afin d'exécuter le Job.

La nouvelle règle a été créée et la valeur de chaque élément de 2009 est *123*.

	A
1	localhost/Biker
2	Orders
3	**All Customers**
4	**All Channels**
5	**All Orders**
6	**All Datatypes**
7	Units
8	**All Products**
9	**All Months**
10	2009
11	
12	123
13	

tPaloRuleList

Propriétés du tPaloRuleList

Famille de composant	Business Intelligence/Cube OLAP/Palo	
Fonction	Le composant **tPaloRuleList** récupère les informations d'une liste de règles d'une base de données Palo spécifiée.	
Objectif	Ce composant liste toutes les règles, les formules, les commentaires, les statuts d'activation, les IDs externes d'un cube donné.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
Connection configuration Indisponible lors de l'utilisation d'une connexion existante.	*Host Name*	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Server Port	Saisissez le numéro du port d'écoute du serveur Palo.
	Username et *Password*	Informations d'authentification de l'utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Database	Saisissez le nom de la base de données dans laquelle se trouve le cube qui vous intéresse.
	Cube	Saisissez le nom du cube dans lequel les informations de règles doivent être retrouvées.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en standalone ou en tant que composant de début d'un processus.	
Connections		Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate. **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Pour plus d'informations à propos des connexions, consultez la section concernant les types de connexions, dans le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_RULES** : nombre de règles. Cette variable est une variable **After** et retourne un nombre entier **EXTERNAL_RULEID** : ID externe de la règle. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Le schéma de sortie est fixe et en lecture seule. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Découvrir le schéma de sortie en lecture seule du tPaloRuleList

Le tableau ci-dessous présente les informations relatives au schéma de sortie en lecture seule du composant**tPaloRuleList**.

Base de données	Type	Description
rule_identifier	long	Identifiant interne de la règle.
rule_definition	string	Nom de la dimension.
Dimension_attribute_cube	string	Formule de la règle. Pour plus d'informations concernant cette règle, consultez le Guide utilisateur Palo.
rule_extern_id	string	ID externe personnalisé.
rule_comment	string	Commentaire utilisateur de la règle.
rule_activated	boolean	Indique si la règle a été activée ou non.

Scénario : Récupérer des informations détaillées concernant des règles d'un cube donné

Le Job de ce scénario récupère les informations concernant les règles appliquées dans les dimensions d'un cube donné.

Pour reproduire ce scénario, procédez comme suit :

Construire le Job

1. Déposez un composant **tPaloRuleList** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Cliquez-droit sur le **tPaloRuleList** afin d'ouvrir son menu contextuel.
3. Dans ce menu, sélectionnez **Row** > **Main** afin de lier les deux composants.

Configurer le composant tPaloRuleList

1. Double-cliquez sur le **tPaloRuleList** pour ouvrir sa vue **Component**.

2. Dans le champ **Host name**, saisissez le nom de l'hôte ou l'adresse IP du serveur hôte, *localhost* dans cet exemple.

3. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo, *7777* dans cet exemple.

4. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.

5. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle se trouvent les dimensions appliquant les règles qui vous intéressent, *Biker* dans cet exemple.

6. Dans le champ **Cube**, saisissez le nom du cube auquel les règles qui vous intéressent appartiennent.

Exécuter le Job

Appuyez sur **F6** pour exécuter le Job.

Les informations de toutes les règles du cube*Orders* sont récupérées et affichées dans la console de la vue **Run**.

```
Starting job tpalo at 16:34 12/11/2010.

[statistics] connecting to socket on port 3743
[statistics] connected
.---------------+---------------+--------------+---------------+-------------.
|                                tLogRow_1                                    |
|=--------------+---------------+--------------+---------------+-------------=|
|rule_identifier|rule_definition|rule_extern_id|rule_comment   |rule_activated|
|=--------------+---------------+--------------+---------------+-------------=|
|1              |['2009'] = 123 |OrderRule1    |Palo Demo Rules|true          |
'---------------+---------------+--------------+---------------+-------------'
[statistics] disconnected
Job tpalo ended at 16:35 12/11/2010. [exit code=0]
```

Pour plus d'informations concernant le schéma de sortie, consultez *Découvrir le schéma de sortie en lecture seule du tPaloRuleList*.

tParAccelSCD

Propriétés du tParAccelSCD

Famille du composant	Databases/ Business Intelligence/ ParAccel	
Fonction	Le composant **tParAccelSCD** reflète et traque les modifications d'une table ParAccel SCD dédiée.	
Objectif	Le **tParAccelSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Sélectionnez dans la liste le pilote à utiliser.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving Mode	Cochez cette case pour améliorer les performances du système.
	Source keys include Null	Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null. *Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.*
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*End date time details*	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tPostgresPlusSCD

Propriétés du tPostgresPlusSCD

Famille de composant	Databases/Business Intelligence/Postgresplus Server	
Fonction	Le composant **tPostgresPlusSCD** reflète et traque les modifications d'une table SCD PostgresPlus dédiée.	
Objectif	Le **tPostgresPlusSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Property type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Liste des versions de la base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving Mode	Cochez cette case pour améliorer les performances du système.
	Source keys include Null	Cochez cette case pour permettre aux colonnes source clés de contenir des valeurs nulles. *Lorsque cette option est cochée, vous devez faire attention à l'unicité des clé(s) source.*
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*End date time details*	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tPostgresPlusSCDELT

Propriétés du tPostgresPlusSCDELT

Famille de composant	Databases/Business Intelligence/ PostgresPlus	
Fonction	Le composant **tPostgresPlusSCDELT** reflète et traque les modifications d'une table PostgresPlus SCD dédiée.	
Objectif	Le **tPostgresPlusSCDELT** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table PostgresPlus SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Source table	Nom de la table contenant les données à filtrer.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois pour que l'opération d'insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire.

		Drop a table if exists and create : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Surrogate Key	Sélectionnez dans la liste une colonne à utiliser comme clé de substitution.
	Creation	Sélectionnez la méthode à utiliser pour générer la clé de substitution. Pour plus d'informations concernant les méthodes de création, consultez *Clés SCD*.
	Source Keys	Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l'unicité des données entrantes.
	Use SCD Type 1 fields	Utilisez le type 1 si vous n'avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.
	Use SCD Type 2 fields	Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications. **Start date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l'une des colonnes d'entrée du schéma comme date de départ (Start Date) dans la table SCD. **End Date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne **End Date** a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l'option **Fixed Year value** et saisir une année fictive. **Log Active Status** : Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut **true** et **false** . Cette colonne permet de repérer facilement le journal actif. **Log versions** : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.
Advanced settings	*Debug mode*	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure	

	mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type **Row Main**.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tPostgresqlSCD

Propriétés du tPostgresqlSCD

Famille de composant	Databases/Business Intelligence/Postgresql Server	
Fonction	Le composant **tPostgresqlSCD** reflète et traque les modifications d'une table SCD Postgresql dédiée.	
Objectif	Le **tPostgresqlSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez dans la liste la version de la base de données que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving Mode	Cochez cette case pour améliorer les performances du système.
	Source keys include Null	Cochez cette case pour permettre aux colonnes source clés de contenir des valeurs nulles. *Lorsque cette option est cochée, vous devez faire attention à l'unicité des clé(s) source.*
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*End date time details*	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tPostgresqlSCDELT

Propriétés du tPostgresqlSCDELT

Famille de composant	Databases/Business Intelligence/ Postgresql	
Fonction	Le composant **tPostgresqlSCDELT** reflète et traque les modifications d'une table Postgresql SCD dédiée.	
Objectif	Le **tPostgresqlSCDELT** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table Postgresql SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez dans la liste la version de la base de données que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Source table	Nom de la table contenant les données à filtrer.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois pour que l'opération d'insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire.

		Drop a table if exists and create : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Surrogate Key	Sélectionnez dans la liste une colonne à utiliser comme clé de substitution.
	Creation	Sélectionnez la méthode à utiliser pour générer la clé de substitution. Pour plus d'informations concernant les méthodes de création, consultez *Clés SCD*.
	Source Keys	Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l'unicité des données entrantes.
	Use SCD Type 1 fields	Utilisez le type 1 si vous n'avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.
	Use SCD Type 2 fields	Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications. **Start date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l'une des colonnes d'entrée du schéma comme date de départ (Start Date) dans la table SCD. **End Date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne **End Date** a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l'option **Fixed Year value** et saisir une année fictive. **Log Active Status** : Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut **true** et **false** . Cette colonne permet de repérer facilement le journal actif. **Log versions** : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.
Advanced settings	*Debug mode*	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement	

	dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type **Row Main**.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tSPSSInput

Propriétés du tSPSSInput

Famille du composant	Business Intelligence	
Fonction	Le composant **tSPSSInput** lit les données de fichiers SPSS (.sav).	
Objectif	Le **tSPSSInput** envoie des données SPSS pour les écrire, dans un autre fichier par exemple.	
Basic settings	*Sync schema*	Cliquez sur ce bouton pour synchroniser le schéma avec les colonnes du fichier SPSS d'entrée.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Pour ce composant, le schéma des métadonnées est directement récupéré à partir du fichier SPSS d'entrée, c'est pourquoi il est en lecture seule. Vous pouvez cliquer sur **Edit schema** pour voir les métadonnées qui ont été récupérées.
	File name	Indiquez le nom ou le chemin d'accès au fichier SPPS à lire.
	Translate labels	Cochez cette case pour traduire les étiquettes (labels) des valeurs stockées. Si vous cochez cette case, vous devrez à nouveau récupérer les métadonnées.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de début. Il requiert un composant de sortie. *Avant de pouvoir utiliser toutes les fonctionnalités des composants SPSS, assurez-vous de suivre les étapes suivantes : -Si vous avez déjà installé SPSS, ajoutez le chemin du répertoire SPSS, qui se présente comme suit : SET PATH=%PATH%;<DR>:\program \SPSS. -Si vous n'avez pas encore installé SPSS, copiez le lib SPSS IO "spssio32.dll" à partir du CD d'installation et copiez-le dans votre répertoire "system32".*	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Oracle fournit deux types de plateformes JVM (32 bits et 64 bits). Par défaut, la JVM utilisée dans un système d'exploitation 64 bits est la version 64 bits de cette JVM. Le fichier JSPSS.dll utilisé par ce composant est compilé dans la version 32 bits de la JVM. Il est donc nécessaire de configurer la JVM 32 bits pour exécuter un Job dans un *Studio Talend*, installé sous un système d'exploitation 64 bits. Pour ce faire, procédez comme suit : 1. Ouvrez le Job dans l'espace de modélisation graphique. 2. Cliquez sur l'onglet **Run** et cliquez sur l'onglet **Advanced settings**. 3. Cochez la case **Use specific JVM arguments** et cliquez sur le bouton **New...**. 4. Dans le champ **Set the VM argument**, saisissez `-d32` et cliquez sur **OK** afin de valider.

Scénario : Afficher le contenu d'un fichier SPSS

Le scénario suivant crée un Job à deux composants qui permet de lire chaque ligne d'un fichier SPSS (.sav) et de les afficher en sortie dans la console de log de la vue **Run**.

Construire le Job

1. A partir de la **Palette**, déposez les composants **tSPSSInput** et **tLogRow** dans l'espace graphique.

2. Cliquez-droit sur le composant **tPSSInput** et connectez-le au **tLogRow** à l'aide d'un lien de type **Main Row**.

Configurer le composant d'entrée

1. Double-cliquez sur le composant **tSPSSInput** pour afficher la vue **Basic settings** et configurer ses paramètres.

2. Cliquez sur le bouton **[...]** à côté du champ **Filename** pour parcourir vos dossiers jusqu'au fichier SPSS que vous voulez lire.

3. Cliquez sur le bouton **[...]** à côté du champ **Sync schema**. Un message apparaît à l'écran pour vous inviter à récupérer le schéma du fichier SPSS sélectionné.

4. Cliquez sur le bouton **Yes** pour fermer le message.

5. Si nécessaire, cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour visualiser la structure de données prédéfinie dans le fichier SPSS source.

6. Cliquez sur le bouton **OK** pour fermer la boîte de dialogue.

Exécuter du Job

Sauvegardez votre Job, et appuyez sur **F6** afin de l'exécuter.

Le fichier SPSS est lu ligne par ligne, et les champs extraits sont affichés dans la console.

```
Starting job SPSS at 10:28 20/01/2010.

[statistics] connecting to socket on port 3802
[statistics] connected
1.0|1.0|57997.0|198522.0
2.0|1.0|57113.0|203200.0
3.0|1.0|54123.0|200744.0
1.0|0.0|63936.0|187791.0
2.0|0.0|64835.0|195714.0
3.0|0.0|66804.0|208239.0
[statistics] disconnected
Job SPSS ended at 10:28 20/01/2010. [exit code=0]
```

Convertir les valeurs stockées

Afin d'effectuer de convertir les valeurs stockées, procédez comme suit :

1. Dans la vue **Basic settings**, cochez la case **Translate label** si vous voulez traduire les étiquettes (labels) des valeurs stockées.

2. Cliquez à nouveau sur le bouton **Sync Schema**, afin de récupérer le schéma après traduction.

   Un message apparaît, vous demandant si vous souhaitez récupérer le schéma du fichier SPSS défini.

3. Cliquez sur **Yes** (Oui) pour fermer le message et procéder à l'étape suivante.

   Un second message s'ouvre et vous demande si vous souhaitez propager les modifications.

4. Cliquez sur **Yes** (Oui) afin de fermer le message et passer à l'étape suivante.

5. Enregistrez le Job puis appuyez sur la touche **F6** pour l'exécuter.

Le fichier SPSS est lu ligne par ligne et les champs extraits sont affichés sur la console de log de la vue **Run**, après conversion des valeurs stockées.

```
Starting job SPSS at 10:32 20/01/2010.

[statistics] connecting to socket on port 3418
[statistics] connected
Under 21|Female|57997.0|198522.0
21-25|Female|57113.0|203200.0
26-30|Female|54123.0|200744.0
Under 21|Male|63936.0|187791.0
21-25|Male|64835.0|195714.0
26-30|Male|66804.0|208239.0
[statistics] disconnected
Job SPSS ended at 10:32 20/01/2010. [exit code=0]
```

tSPSSOutput

Propriétés du tSPSSOutput

Famille du composant	Business Intelligence	
Fonction	Le composant **tSPSSOutput** écrit des données dans un fichier SPSS (.sav).	
Objectif	Le **tSPSSOutput** écrit ou ajoute des données à un fichier SPSS. Il crée les fichiers SPSS directement ou écrase ceux qui existent déjà.	
Basic settings	*Sync schema*	Cliquez sur ce bouton pour synchroniser le schéma avec les colonnes du fichier SPSS d'entrée.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Filename	Nom ou chemin d'accès du fichier SPSS à à écrire.
	Write Type	Sélectionnez dans la liste le type d'action à effectuer : **Write**: écrit simplement les nouvelles données. **Append**: écrit les nouvelles données à la suite de celles existantes.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant ne peut être utilisé comme composant de début. Il nécessite un flux d'entrée. *Avant de pouvoir utiliser toutes les fonctionnalités des composants SPSS, assurez-vous de suivre les étapes suivantes : -Si vous avez déjà installé SPSS, ajoutez le chemin du répertoire SPSS, qui se présente comme suit : SET PATH=%PATH%;<DR>:\program \SPSS. -Si vous n'avez pas encore installé SPSS, copiez le lib SPSS IO "spssio32.dll" à partir du CD d'installation et copiez-le dans votre répertoire "system32".*
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Oracle fournit deux types de plateformes JVM (32 bits et 64 bits). Par défaut, la JVM utilisée dans un système d'exploitation 64 bits est la version 64 bits de cette JVM. Le fichier JSPSS.dll utilisé par ce composant est compilé dans la version 32 bits de la JVM. Il est donc nécessaire de configurer la JVM 32 bits pour exécuter un Job dans un *Studio Talend*, installé sous un système d'exploitation 64 bits. Pour ce faire, procédez comme suit : 1. Ouvrez le Job dans l'espace de modélisation graphique. 2. Cliquez sur l'onglet **Run** et cliquez sur l'onglet **Advanced settings**. 3. Cochez la case **Use specific JVM arguments** et cliquez sur le bouton **New...**. 4. Dans le champ **Set the VM argument**, saisissez `-d32` et cliquez sur **OK** afin de valider.

Scénario : Ecrire des données dans un fichier .sav

Ce scénario décrit un Job très simple qui écrit des données d'entrée dans un fichier .sav.

Construire le Job

1. Glissez-déposez de la **Palette** dans l'espace de modélisation graphique un **tRowGenerator** et un **tSPSSOutput**.
2. Cliquez-droit sur le **tRowGenerator**, et connectez-le au **tSPSSOutput**, à l'aide d'un lien **Row Main**.

Configurer le composant d'entrée

1. Dans l'espace de modélisation graphique, double-cliquez sur le **tRowGenerator** afin d'afficher sa vue **Basic settings** et ouvrir son éditeur. Vous pouvez définir votre schéma.

2. Cliquez sur le bouton **[+]** afin d'ajouter les colonnes que vous voulez écrire dans le fichier .sav.

3. Définissez le schéma, et configurez les paramètres des colonnes.

   Vérifiez que vous avez défini la longueur de vos colonnes. Dans le cas contraire, un message d'erreur s'affichera lors de la construction du Job.

4. Cliquez sur **OK** afin de valider votre schéma et fermer l'éditeur.

Configurer le composant de sortie

1. Double-cliquez sur le **tSPSSOutput** afin d'afficher sa vue **Component** et définir ses propriétés.

2. Cliquez sur le bouton **[...]** à côté du champ **Filename**, et parcourez votre répertoire jusqu'au fichier SPSS .sav dans lequel vous souhaitez écrire les données.

3. Cliquez sur le bouton **[...]** à côté du bouton **Sync columns**, afin de synchroniser les colonnes avec celles du composant précédent. Dans cet exemple, le schéma devant être inséré dans le fichier .sav est composé de deux colonnes : *id* et *country*.

4. Si nécessaire, cliquez sur le bouton **Edit schema** pour voir/éditer le schéma défini.

5. Dans la liste **Write Type**, sélectionnez **Write** (écrire) ou **Append** (écrire à la suite), pour simplement écrire les données d'entrée dans le fichier .sav, ou bien les écrire à la fin de ce même fichier.

Exécuter le Job

Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

Les données générées par le **tRowGenerator** sont écrites dans le fichier .sav défini.

tSPSSProperties

Propriétés du tSPSSProperties

Famille du composant	Business Intelligence	
Fonction	Le composant **tSPSSProperties** décrit les propriétés d'un fichier SPSS (.sav) défini.	
Objectif	Le **tSPSSProperties** permet d'obtenir des informations sur les propriétés principales d'un fichier SPSS défini.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Pour ce composant, le schéma des métadonnées est prédéfini, c'est pourquoi il est en lecture seule. Il correspond à la convention interne SPSS. Vous pouvez cliquer sur **Edit schema** pour voir les métadonnées prédéfinies.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Filename	Nom ou chemin d'accès au fichier SPSS à traiter.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**IS_VALID_FILE** : résultat de la validation du fichier. Cette variable est une variable **After** et retourne un booléen. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de début. Il requiert un composant de sortie. *Pour pouvoir utiliser toutes les fonctionnalités des composants SPSS, assurez-vous de suivre les étapes suivantes : -Si vous avez déjà installé SPSS, ajoutez le chemin du répertoire SPSS, qui se présente comme suit : SET PATH=%PATH%;<DR>:\program \SPSS. -Si vous n'avez pas encore installé SPSS, copiez le lib SPSS IO "spssio32.dll" à partir du CD d'installation et copiez-le dans votre répertoire "system32".*	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Oracle fournit deux types de plateformes JVM (32 bits et 64 bits). Par défaut, la JVM utilisée dans un système d'exploitation 64 bits est la version 64 bits de cette JVM. Le fichier JSPSS.dll utilisé par ce composant est compilé dans la version 32 bits de la JVM. Il est donc nécessaire de configurer la JVM 32 bits pour exécuter un Job dans un *Studio Talend*, installé sous un système d'exploitation 64 bits. Pour ce faire, procédez comme suit : 1. Ouvrez le Job dans l'espace de modélisation graphique. 2. Cliquez sur l'onglet **Run** et cliquez sur l'onglet **Advanced settings**. 3. Cochez la case **Use specific JVM arguments** et cliquez sur le bouton **New...**. 4. Dans le champ **Set the VM argument**, saisissez `-d32` et cliquez sur **OK** afin de valider.

Scénario associé

Pour des scénarios associés, consultez :

- *Scénario : Lire les données maître d'un hub MDM*
- *Scénario : Ecrire des données dans un fichier .sav*

tSPSSStructure

Propriétés du tSPSSStructure

Famille du composant	Business Intelligence	
Fonction	Le composant **tSPSSStructure** récupère les informations sur les variables contenues dans le fichier SPSS (.sav).	
Objectif	Le **tSPSSStructure** traite les variables contenues dans des fichiers SPSS. Vous pouvez utiliser ce composant avec le **tFileList** pour récupérer les informations des fichiers *.sav existant et les utiliser pour analyser de manière plus approfondie ou pour vérifier vos résultats.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Pour ce composant, le schéma des métadonnées est prédéfini, c'est pourquoi il est en lecture seule. Il correspond à la convention interne SPSS. Vous pouvez cliquer sur **Edit schema** pour voir les métadonnées prédéfinies.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Filename	Nom ou chemin d'accès au fichier SPSS à traiter.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de début. Il requiert un composant de sortie. *Avant de pouvoir utiliser toutes les fonctionnalités des composants SPSS, assurez-vous de suivre les étapes suivantes : -Si vous avez déjà installé SPSS, ajoutez le chemin du répertoire SPSS, qui se présente comme suit : SET PATH=%PATH%;<DR>:\program \SPSS. -Si vous n'avez pas encore installé SPSS, copiez le lib SPSS IO "spssio32.dll" à partir du CD d'installation et copiez-le dans votre répertoire "system32".*	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Oracle fournit deux types de plateformes JVM (32 bits et 64 bits). Par défaut, la JVM utilisée dans un système d'exploitation 64 bits est la version 64 bits de cette JVM. Le fichier JSPSS.dll utilisé par ce composant est compilé dans la version 32 bits de la JVM. Il est donc nécessaire de configurer la JVM 32 bits pour exécuter un Job dans un *Studio Talend*, installé sous un système d'exploitation 64 bits. Pour ce faire, procédez comme suit : 1. Ouvrez le Job dans l'espace de modélisation graphique. 2. Cliquez sur l'onglet **Run** et cliquez sur l'onglet **Advanced settings**. 3. Cochez la case **Use specific JVM arguments** et cliquez sur le bouton **New...**. 4. Dans le champ **Set the VM argument**, saisissez `-d32` et cliquez sur **OK** afin de valider.

Scénario associé

Pour des scénarios associés, consultez :

- *Scénario : Lire les données maître d'un hub MDM*.
- *Scénario : Ecrire des données dans un fichier .sav*.

tSybaseSCD

Propriétés du tSybaseSCD

Famille du composant	Databases/Business Intelligence/Sybase	
Fonction	Le composant **tSybaseSCD** reflète et traque les modifications d'une table SCD Sybase dédiée.	
Objectif	**Le tSybaseSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving Mode	Cochez cette case pour améliorer les performances du système.
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	End date time details	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tSybaseSCDELT

Propriétés du tSybaseSCDELT

Famille du composant	Databases/ Business Intelligence/ Sybase	
Fonction	Le composant **tSybaseSCDELT** reflète et traque les modifications d'une table Sybase SCD dédiée.	
Objectif	Le **tSybaselSCDELT** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table Sybase SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Source table	Nom de la table contenant les données à filtrer.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois pour que l'opération d'insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Clear a table** : supprimer le contenu de la table.

		Truncate table : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Surrogate Key	Sélectionnez dans la liste une colonne à utiliser comme clé de substitution.
	Creation	Sélectionnez la méthode à utiliser pour générer la clé de substitution. Pour plus d'informations concernant les méthodes de création, consultez *Clés SCD*.
	Source Key	Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l'unicité des données entrantes.
	Use SCD Type 1 fields	Utilisez le type 1 si vous n'avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.
	Use SCD Type 2 fields	Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications. **Start date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l'une des colonnes d'entrée du schéma comme date de départ (Start Date) dans la table SCD. **End Date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne **End Date** a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l'option **Fixed Year value** et saisir une année fictive. **Log Active Status** : Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut **true** et **false**. Cette colonne permet de repérer facilement le journal actif. **Log versions** : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	Debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant.	

	La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type **Row Main**.
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

tVerticaSCD

Propriétés du tVerticaSCD

Famille de composant	Databases/ Vertica	
Fonction	Le composant **tVerticaSCD** reflète et traque les modifications d'une table Vertica SCD dédiée.	
Objectif	Le **tVerticaSCD** répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version de Vertica que vous utilisez.
	Host	Saisissez l'adresse IP du serveur de base de données.
	Port	Saisissez le numéro du port d'écoute du serveur de la base de données.
	Database	Saisissez le nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Une seule table peut être écrite à le fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : aucune opération n'est effectuée. **Create a table** : la table n'existe pas et est créée. **Create a table if not exists** : la table est créée si elle n'existe pas.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.

		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	SCD Editor	L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d'informations, consultez *Méthodologie de gestion du SCD*.
	Use memory saving mode	Cochez cette case pour améliorer les performances du système.
	Die on error	Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Additional JDBC Parameters*	Ajoutez des paramètres JDBC supplémentaires nécessaires à la connexion à la base de données. Cette option n'est pas disponible lorsque vous cochez la case **Use an existing connection** dans la vue **Basic settings**.
	End date time details	Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*. Ce champ apparaît uniquement lorsqu'un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD. Pour plus d'informations concernant le SCD de **Type 2**, consultez *Méthodologie de gestion du SCD*.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
	Debug mode	Cochez cette case pour afficher chaque étape durant le traitement des entrées dans une base de données.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant de sortie. Il nécessite un composant d'entrée et un lien **Row** > **Main** en entrée.

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénario associé

Pour un scénario associé, consultez *tMysqlSCD*.

Composants Cloud

Ce chapitre passe en revue les principaux composants présents dans la famille **Cloud** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

Les bases de données, les services de données et les applications basées SaaS (CRM, HR, ERP, etc.) hébergés dans des clouds privés et publics se multiplient, accompagnés d'applications et de bases de données « sur site » qui constituent le fondement de l'informatique d'entreprise

La famille **Cloud** regroupe les connecteurs de bases de données les plus populaires. Ces connecteurs couvrent divers besoins : ouverture de connexion, lecture et écriture de tables, commit de transactions, ainsi que des rollbacks pour le traitement des erreurs.

tAmazonAuroraClose

Propriétés du tAmazonAuroraClose

Famille de composant	Cloud/AmazonRDS/Aurora	
Fonction	Le composant **tAmazonAuroraClose** ferme une connexion active à une instance de la base de données Amazon Aurora.	
Objectif	Ce composant vous permet de fermer une connexion active à une instance d'une base de données Amazon Aurora afin de libérer des ressources.	
Basic settings	*Component List*	Sélectionnez dans la liste le composant **tAmazonAuroraConnection** ouvrant la connexion à fermer.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Amazon Aurora, notamment avec le **tAmazonAuroraConnection** et le **tAmazonAuroraCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

La base de données Aurora est compatible avec MySQL v5.6. Pour un scénario similaire, consultez :

- *Scénario 3 : Lire des données dans des bases de données MySQL à l'aide de connexions dynamiques basées sur les variables de contexte*.

tAmazonAuroraCommit

Propriétés du tAmazonAuroraCommit

Famille de composant	Cloud/AmazonRDS/Aurora	
Fonction	Le composant **tAmazonAuroraCommit** valide les données traitées par le Job dans la base de données connectée Amazon Aurora.	
Objectif	A l'aide d'une connexion unique, ce composant commite en une fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component List*	Sélectionnez le composant **tAmazonAuroraConnection** pour lequel vous souhaitez commiter l'action à effectuer.
	Close Connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tAmazonAuroraCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Amazon Aurora, notamment avec le **tAmazonAuroraConnection** et le **tAmazonAuroraRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

La base de données Aurora est compatible avec MySQL v5.6. Pour un scénario similaire, consultez :

- *Scénario 1 : Insérer des données dans des tables mère/fille*

tAmazonAuroraConnection

Propriétés du tAmazonAuroraConnection

Famille de composant	Cloud/AmazonRDS/Aurora	
Fonction	Le composant **tAmazonAuroraConnection** ouvre une connexion à une instance de la base de données Amazon Aurora.	
Objectif	Ce composant vous permet d'ouvrir une connexion à une instance de la base de données pouvant être réutilisée par d'autres composants Amazon Aurora.	
Basic settings	*Property Type*	Peut être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Saisissez l'adresse IP ou le nom de l'hôte de la base de données Amazon Aurora.
	Port	Saisissez le numéro du port d'écoute de la base de données Amazon Aurora.
	Database	Saisissez le nom de la base de données à utiliser.
	Additional JDBC parameters	Spécifiez des informations supplémentaires de connexion à la base de données créée.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* Cette case est indisponible lorsque la case **Specify a data source alias** est cochée.
	Shared DB Connection Name	Saisissez le nom de la connexion partagée. Ce champ s'affiche uniquement lorsque la case **Use or register a shared DB Connection** est cochée.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. Cette case est indisponible lorsque la case **Use or register a shared DB Connection** est cochée.
	Data source alias	Saisissez l'alias de la source de données créée, du côté *Talend Runtime*.

		Ce champ s'affiche lorsque la case **Specify a data source alias** est cochée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Amazon Aurora, notamment avec le **tAmazonAuroraCommit** et le **tAmazonAuroraRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

La base de données Aurora est compatible avec MySQL v5.6. Pour un scénario similaire, consultez :

- *Scénario 1 : Insérer des données dans des tables mère/fille*.
- *Scénario 2 : Partager une connexion de base de données entre un Job père et un Job fils*.

tAmazonAuroraInput

Propriétés du tAmazonAuroraInput

Famille de composant	Cloud/AmazonRDS/Aurora	
Fonction	Le composant **tAmazonAuroraInput** lit une base de données Amazon Aurora et en extrait des champs à l'aide de requêtes.	
Objectif	Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux **Row** > **Main**.	
Basic settings	*Property Type*	Peut être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Saisissez l'adresse IP ou le nom de l'hôte de la base de données Amazon Aurora.
	Port	Saisissez le numéro du port d'écoute de la base de données Amazon Aurora.
	Database	Saisissez le nom de la base de données à utiliser.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Saisissez le nom de la table à lire.
	Query Type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Guess Query	Cliquez sur le bouton afin de générer la requête correspondant au schéma de la table dans le champ **Query**.
	Guess schema	Cliquez sur ce bouton afin de récupérer le schéma de la table.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. Cette case est indisponible lorsque la case **Use an existing connection** est cochée.
	Data source alias	Saisissez l'alias de la source de données créée du côté *Talend Runtime*. Ce champ s'affiche uniquement lorsque la case **Specify a data source alias** est cochée.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Lorsque vous devez gérer des données de type *0000-00-00 00:00:00* à l'aide de ce composant, configurez le paramètre à `noDatetimeStringSync=true&zeroDateTimeBehavior=convertToNull`. Ce champ est indisponible lorsque la case **Use an existing connection** est cochée dans l'onglet **Basic settings**.
	Enable stream	Cochez cette case pour préférer le streaming à la mise en mémoire tampon, ce qui permet au code de lire les données d'une table volumineuse sans consommer beaucoup de mémoire, afin d'optimiser les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Cochez la (les) case(s) de la colonne **Trim** afin de supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées. L'option est indisponible lorsque la case **Trim all the String/Char columns** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**.	

	Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en tant que composant de début dans un Job ou sous-job et nécessite un lien de sortie.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

La base de données Aurora est compatible avec MySQL v5.6. Pour un scénario similaire, consultez :

- *Scénario 1 : Ecrire des colonnes d'une base de données MySQL dans un fichier de sortie*.
- *Scénario 3 : Lire des données dans des bases de données MySQL à l'aide de connexions dynamiques basées sur les variables de contexte*.

tAmazonAuroraOutput

Propriétés du tAmazonAuroraOutput

Famille de composant	Cloud/AmazonRDS/Aurora	
Fonction	Le composant **tAmazonAuroraOutput** écrit, met à jour, modifie ou supprime des entrées dans une base de données Amazon Aurora.	
Objectif	Ce composant exéctue l'action définie sur la table et/ou sur les données contenues dans la table, en fonction du flux entrant du composant précédent dans le Job.	
Basic settings	*Property Type*	Peut être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Saisissez l'adresse IP ou le nom de l'hôte de la base de données Amazon Aurora.
	Port	Saisissez le numéro du port d'écoute de la base de données Amazon Aurora.
	Database	Saisissez le nom de la base de données à utiliser.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur à la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Saisissez le nom de la table à écrire. Notez qu'une table peut être écrite à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : • **None** : n'effectuer aucune opération sur la table. • **Drop and create table** : supprimer la table puis en créer une nouvelle. • **Create table** : créer une table qui n'existe pas encore. • **Create table if not exists** : créer la table si nécessaire.

		• **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. • **Clear table** : supprimer le contenu de la table. • **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : • **Insert** : ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. • **Update** : mettre à jour les entrées existantes. • **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. • **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. • **Delete** : supprimer les entrées correspondant au flux d'entrée. • **Replace**: ajouter de nouvelles entrées à la table. Si une ancienne ligne dans la table a la même valeur qu'une nouvelle ligne pour un index PRIMARY KEY ou UNIQUE, l'ancienne ligne est supprimée avant insertion de la nouvelle. • **Insert or update on duplicate key or unique index** : ajouter des entrées si la valeur insérée n'existe pas ou mettre à jour les entrées si la valeur insérée existe déjà et qu'un risque de violation d'une clé unique se présente. • **Insert ignore** : ajouter uniquement de nouvelles lignes afin d'empêcher les erreurs de doublons de clés. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. Cette case est indisponible lorsque la case **Use an existing connection** est cochée.
	Data source alias	Saisissez l'alias de la source de données créée du côté *Talend Runtime*. Ce champ s'affiche uniquement lorsque la case **Specify a data source alias** est cochée.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Extend Insert	Cochez cette case pour insérer un ensemble de lignes définies au lieu d'insérer les lignes une par une. Cette option permet un important gain de performance. Cette case est disponible lorsque l'option **Insert** est sélectionnée dans la liste **Action on data**, dans l'onglet **Basic settings**. Cette option n'est pas compatible avec le lien **Reject**. Vous devez donc la décocher si vous utilisez un lien **Reject** en sortie du composant.
	Number of rows per insert	Saisissez le nombre de lignes à insérer par opération. Notez que, plus haute est la valeur spécifiée, plus la performance est basse, à cause de la consommation de mémoire. Ce champ est disponible lorsque la case **Extend Insert** est cochée.
	Use Batch Size	Cochez cette case afin d'activer le mode batch pour le traitement des données. Cette case est disponible uniquement si vous avez choisi l'option **Update** ou **Delete** dans la liste **Action on data** de la vue **Basic settings**.
	Batch Size	Saisissez le nombre de lignes à traiter dans chaque lot. Ce champ est disponible lorsque la case **Use Batch Size** est cochée.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier. Cliquez sur le bouton **[+]** sous la table pour ajouter des colonnes et configurez les paramètres suivants pour chaque colonne. • **Name** : saisissez le nom de la colonne du schéma à modifier ou insérer. • **SQL expression** : saisissez l'instruction SQL à exécuter afin de modifier ou insérer les données dans la colonne correspondante. • **Position** : sélectionnez **Before**, **After** ou **Replace**, selon l'action à effectuer sur la colonne de référence.

		• **Reference column** : saisissez une colonne de référence pouvant être utilisée par le **tAmazonAuroraOutput** pour localiser ou remplacer la nouvelle colonne ou la colonne à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données. • **Key in update** : cochez la case de la colonne correspondante, par rapport à laquelle les données sont mises à jour. • **Key in delete** : cochez la case de la colonne correspondante, par rapport à laquelle les données sont supprimées. • **Updatable** : cochez la case si les données de la colonne correspondante peuvent être mises à jour. • **Insertable** : cochez la case si les données de la colonne correspondante peuvent être insérées.
	Use Hint Options	Cochez cette case pour configurer les indicateurs pouvant permettre d'optimiser l'exécution d'une requête.
	Hint Options	Cochez cette case pour activer la zone de configuration des indicateurs (ou Hints) permettant d'optimiser l'exécution d'une requête. Dans cette zone, les paramètres sont : • **HINT** : spécifiez l'indicateur dont vous avez besoin, en utilisant la syntaxe `/* + */`. • **POSITION** : spécifiez la place de l'indicateur dans une instruction SQL. • **SQL STMT***: sélectionnez l'instruction SQL **INSERT**, **UPDATE**, ou **DELETE** à utiliser.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	Use duplicate key update mode insert	Cochez cette case pour activer le mode ON DUPLICATE KEY UPDATE, puis cliquez sur le bouton **[+]** sous la table pour ajouter des colonnes à mettre à jour et spécifiez l'action de mise à jour à effectuer sur la colonne correspondante. • **Column** : saisissez le nom de la colonne à mettre à jour. • **Value** : saisissez l'action à effectuer sur la colonne. Cette case est disponible lorsque l'option **Insert** est sélectionnée dans la liste **Action on data** de la vue **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier.	

	NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé comme composant de sortie. Il vous permet d'effectuer des actions sur les données d'une table dans une base de données Amazon Aurora. Il permet également de créer un flux de rejet, à l'aide d'un lien **Row** > **Rejects** afin de filtrer les données en erreur. Pour un scénario similaire, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

La base de données Aurora est compatible avec MySQL v5.6. Pour des scénarios associés, consultez :

- *Scénario 1 : Ajouter une colonne et modifier les données*
- *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject*

tAmazonAuroraRollback

Propriétés du tAmazonAuroraRollback

Famille de composant	Cloud/AmazonRDS/Aurora	
Fonction	Le composant **tAmazonAuroraRollback** annule la transaction dans une base de données connectée Amazon Aurora.	
Objectif	Ce composant vous permet d'annuler des modifications apportées à la base de données Amazon Aurora pour éviter le commit de transaction involontaire si une erreur survient.	
Basic settings	*Component List*	Sélectionnez le composant **tAmazonAuroraConnection** pour lequel effectuer l'action de rollback.
	Close Connection	Cette case est cochée par défaut est vous permet de fermer la connexion à la base de données une fois le rollback effectué. Décochez cette case pour continuer à utiliser la connexion sélectionnée une fois sa tâche terminée.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Amazon Aurora, notamment avec le **tAmazonAuroraConnection** et le **tAmazonAuroraCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

La base de données Aurora est compatible avec MySQL v5.6. Pour un scénario similaire, consultez :

- *Scénario : Annuler l'insertion de données dans des tables mère/fille*

tAmazonMysqlClose

Propriétés du tAmazonMysqlClose

Fonction	Le composant **tAmazonMysqlClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tAmazonMysqlConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être utilisé avec des composants **AmazonMysql**, notamment avec le **tAmazonMysqlCommit** et le **tAmazonMysqlConnection**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tAmazonMysqlConnection** et **tAmazonMysqlRollback**. Il est généralement utilisé avec un composant **tAmazonMysqlConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tAmazonMysqlClose**, consultez *tMysqlConnection*.

tAmazonMysqlCommit

Propriétés du tAmazonMysqlCommit

Famille de composant	Cloud/AmazonRDS/MySQL	
Fonction	Le composant **AmazontMysqlCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tAmazonMysqlConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tAmazonMysqlCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants AmazonMysql et notamment avec les composants **tAmazonMysqlConnection** et **tAmazonMysqlRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé au composant **tAmazonMysqlCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tAmazonMysqlConnection

Propriétés du tAmazonMysqlConnection

Famille de composant	Cloud/AmazonRDS/MySQL	
Fonction	Le composant **tAmazonMysqlConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	MySQL 5 est disponible.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Additional JDBC parameters	Spécifiez des informations supplémentaires de connexion à la base de données créée.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications

		doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants AmazonMysql, notamment avec les composants **tAmazonMysqlCommit** et **tAmazonMysqlRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario utilisant ce composant, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*

tAmazonMysqlInput

Propriétés du tAmazonMysqlInput

Famille de composant	Cloud/AmazonRDS/MySQL	
Fonction	Le composant **tAmazonMysqlInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tAmazonMysqlInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	MySQL 5 est disponible.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**) , soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Table Name	Nom de la table à lire.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Lorsque vous devez traiter des données au format date/heure *0000-00-00 00:00:00* utilisant ce composant, définissez les paramètres comme suit : `noDatetimeStringSync=true&zero-DateTimeBehavior=convertToNull`.
	Enable stream	Cochez cette case pour déterminer avec quelles lignes vous souhaitez travailler. Cette option permet d'améliorer les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées. Décochez **Trim all the String/Char columns** pour activer le tableau **Trim column**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données MySQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour des scénarios associés, consultez *tMysqlInput*.

tAmazonMysqlOutput

Propriétés du tAmazonMysqlOutput

Famille de composant	Cloud/AmazonRDS/MySQL	
Fonction	Le composant **tAmazonMysqlOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tAmazonMysqlOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	MySQL 5 est disponible.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **Default** : n'effectuer aucune opération sur la table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire.

		Drop a table if exists and create : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. **Insert or update on duplicate key or unique index** : Ajouter des entrées si la valeur insérée n'existe pas ou mettre à jour les entrées si la valeur insérée existe déjà et qu'un risque de violation d'une clé unique se présente. **Insert ignore** : Ajouter uniquement de nouvelles lignes afin d'empêcher les erreurs de doublons de clés. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**) , soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.

		Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Extend Insert	Cochez cette case pour insérer un ensemble de lignes définies au lieu d'insérer les lignes une par une. Cette option permet un important gain de performance. **Number of rows per insert** : saisissez le nombre de lignes à insérer en un bloc. Notez que si vous sélectionnez un nombre important de lignes, cela peut augmenter la quantité de mémoire utilisée et donc diminuer les performances. Cette option n'est pas compatible avec le lien **Reject**. Vous devez donc la décocher si vous utilisez un lien **Reject** en sortie du composant. *Si vous utilisez ce composant avec un **tMysqlLastInsertID**, soyez sûr de décocher la case **Extend Insert** dans la vue **Advanced settings**. **Extend Insert** permet le chargement par lots, cependant, si la case est cochée, seul l'ID de la dernière ligne du dernier lot sera retourné.*
	Use Batch size	Cochez cette case afin d'activer le mode batch pour le traitement des données. Dans le champ **Batch Size** qui apparaît, vous pouvez saisir le nombre de lignes par lot à charger. Cette case est disponible uniquement si vous avez choisi l'option **Insert**, **Update**, **Single Insert Query** ou **Delete** dans le champ **Action on data**.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tAmazonMysqlOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Use Hint Options	Cochez cette case pour activer la zone de configuration des indicateurs (ou Hints) permettant d'optimiser l'exécution d'une requête. Dans cette zone, les paramètres sont : - **HINT** : spécifiez l'indicateur dont vous avez besoin, en utilisant la syntaxe `/*+ */` - **POSITION** : spécifiez la place de l'indicateur dans une instruction SQL. - **SQL STMT** : sélectionnez l'instruction SQL que vous souhaitez utiliser.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	Use duplicate key update mode insert	Met à jour les valeurs des colonnes spécifiées en cas de doublon de la clé primaire. **Column** : Saisissez entre guillemets le nom de la colonne à mettre à jour. **Value** : Saisissez l'opération que vous souhaitez effectuer sur la colonne. Pour utiliser cette option, vous devez préalablement sélectionner le mode **Insert** dans la liste **Action on data** de la vue **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.

Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Ce composant doit être utilisé en tant que composant de sortie. Il permet de faire des actions sur une table ou les données d'une table d'une base de données MySQL. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation du **tAmazonMysqlOutput**, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour des scénarios associés, consultez *tMysqlSCD*.

tAmazonMysqlRollback

Propriétés du tAmazonMysqlRollback

Famille de composant	Cloud/AmazonRDS/Mysql	
Fonction	Le composant **tAmazonMysqlRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tAmazonMysqlConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants AmazonMySQL, notamment les composants **tAmazonMysqlConnection** et **tAmazonMysqlCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille*.

tAmazonMysqlRow

Propriétés du tAmazonMysqlRow

Famille de composant	Cloud/Amazon/MySQL	
Fonction	Le composant **tAmazonMysqlRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, le **tAmazonMysqlRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	MySQL 5 est disponible.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

		Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
	Table Name	Nom de la table à traiter.
	Query type	Peut être Built-in ou Repository.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*

tAmazonOracleClose

Propriétés du tAmazonOracleClose

Fonction	Le composant **tAmazonOracleClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données AmazonOracle.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tAmazonOracleConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants AmazonOracle, notamment **tAmazonOracleConnection** et **tAmazonOracleCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tAmazonOracleConnection** et **tAmazonOracleRollback**. Il est généralement utilisé avec un composant **tAmazonOracleConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tAmazonOracleClose**, consultez *tMysqlConnection*.

tAmazonOracleCommit

Propriétés du tAmazonOracleCommit

Famille de composant	Cloud/AmazonRDS/Oracle	
Fonction	Le composant **tAmazonOracleCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tAmazonOracleConnection** dans la liste.
	Close Connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tOracleCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants AmazonOracle et notamment **tAmazonOracleConnection** et **tAmazonOracleRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé au composant **tAmazonOracleCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tAmazonOracleConnection

Propriétés du tAmazonOracleConnection

Famille de composant	Cloud/AmazonRDS/Oracle	
Fonction	Le composant **tAmazonOracleConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Connection type	Liste déroulante des pilotes disponibles : **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.
	DB Version	Oracle 11-5 est disponible.
	Use tns file	Cochez cette case pour utiliser les métadonnées d'un contexte contenu dans un fichier tns. Un même fichier Tns peut contenir plusieurs contextes. **TNS File** : Renseignez manuellement le chemin d'accès ou cliquez sur le bouton **[...]** pour parcourir vos dossiers jusqu'au fichier Tns à utiliser. **Select a DB Connection in Tns File** : Cliquez sur le bouton **[...]** pour afficher tous les contextes du fichier Tns et sélectionner celui à utiliser.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Additional JDBC parameters	Spécifiez des informations supplémentaires de connexion à la base de données créée. Dans ce champ, vous pouvez paramétrer le type d'encodage.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant.

		*Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants AmazonOracle, notamment les composants **tAmazonOracleCommit** et **tAmazonOracleRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tAmazonOracleConnection**, consultez *tMysqlConnection*.

tAmazonOracleInput

Propriétés du tAmazonOracleInput

Famille de composant	Cloud/AmazonRDS/Oracle	
Fonction	Le composant **tAmazonOracleInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tAmazonOracleInput** exécute une requête de base de données dans laquelle l'ordre doit strictement correspondre à l'ordre défini dans le schéma. Puis la liste des champs est transmise au composant via une connexion Main Row.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionne le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Connection type	Liste déroulante des pilotes disponibles : **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur in système.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Oracle schema	Nom du schéma Oracle.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**

		Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in: Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository: Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Table name	Nom de la table de base de données.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
	Use cursor	Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d'optimiser les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données AmazonOracle.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour des scénarios associés, consultez :

- Le premier scénario du composant **tDBInput**, dans *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- Le second scénario du composant **tDBInput**, dans *Scénario 2 : Utiliser la variable StoreSQLQuery*.
- Le scénario du composant **tContextLoad**, dans *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement*.

tAmazonOracleOutput

Propriétés du tAmazonOracleOutput

Famille de composant	Cloud/AmazonRDS/Oracle	
Fonction	Le composant **tAmazonOracleOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tAmazonOracleOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Liste déroulante des pilotes disponibles : **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Oracle schema	Nom du schéma Oracle.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	La liste **Action on table** n'est pas disponible si vous cochez la case **Enable parallel execution** dans la vue **Advanced settings**.

		Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. *Si vous cochez la case **Use an existing connection** et sélectionnez une option différente de **None** dans la liste **Action on table**, une instruction de commit est générée automatiquement avant l'opération de mise à jour/insertion/suppression de données.*
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondant au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).

	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Override any existing NLS_LANG environment variable	Cochez cette case pour écraser les variables d'environnement NLS_LANG déjà définies.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tOracleOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Use Hint Options	Cochez cette case afin d'activer la configuration de la zone des Hints, vous permettant ainsi d'optimiser l'exécution d'une requête. Dans cette zone, les paramètres sont : - **HINT** : spécifiez l'hint dont vous avez besoin, à l'aide de la syntaxe `/*+ */`. -**POSITION** : spécifiez où mettre l'hint dans une instruction SQL. - **SQL STMT** : sélectionnez l'instruction SQL que vous souhaitez utiliser.
	Convert columns and table to uppercase	Cochez cette case pour passer les noms de colonnes et de tables en majuscule.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Use Batch Size	Nombre de lignes dans chaque lot de traitement de données. *Si cette case n'apparaît pas, vérifiez que la case **Use an existing connection** dans les **Basic settings** est décochée.*
	Support null in "SQL WHERE" statement	Cochez cette case pour prendre en compte les valeurs Null lors d'une instruction WHERE.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**.	

	Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Ce composant doit être utilisé en tant que composant de sortie. Il permet de faire des actions sur une table ou les données d'une table d'une base de données AmazonOracle. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour des scénarios associés au composant **tAmazonOracleOutput**, consultez :

- **tDBOutput :** dans *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC*.
- **tMysqlOutput :** dans *Scénario 1 : Ajouter une colonne et modifier les données*.

tAmazonOracleRollback

Propriétés du tAmazonOracleRollback

Famille de composant	Cloud/AmazonRDS/Oracle	
Fonction	Le composant **tAmazonOracleRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **AmazontOracleConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants AmazonOracle, notamment les composants **tAmazonOracleConnection** et **tAmazonOracleCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tAmazonOracleRollback**, consultez *tMysqlRollback*.

tAmazonOracleRow

Propriétés du tAmazonOracleRow

Famille de composant	Cloud/AmazonRDS/Oracle	
Fonction	Le composant **tAmazonOracleRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, le **tAmazonOracleRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Liste déroulante des pilotes disponibles.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Query type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	*Use NB_LINE_*	Cette option vous permet d'alimenter la variable afin de passer au sous-job ou au composant suivant le nombre de lignes mises à jour/supprimées/insérées. Ce champ s'applique uniquement si la requête saisie dans le champ **Query** est une requête INSERT, UPDATE ou DELETE. • **NONE** : n'alimente pas la variable. • **INSERTED** : alimente la variable avec le nombre de lignes insérées. • **UPDATED** : alimente la variable avec le nombre de lignes mises à jour. • **DELETED** : alimente la variable avec le nombre de lignes supprimées.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**.	

	Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- **tDBSQLRow**: *Scénario : Réinitialiser l'auto-incrémentation d'une base de données*.
- **tMySQLRow**: *Scénario 1 : Supprimer et re-générer un index de table MySQL*.

tBoxConnection

Propriétés du tBoxConnection

Famille de composant	Cloud/Box	
Fonction	Ce composant crée une connexion à un compte Box donné.	
Objectif	Le composant **tBoxConnection** crée une connexion à Box que les autre composants Box peuvent réutiliser.	
Basic settings	*Client Key*	Saisissez la clé client requise par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Client Secret	Saisissez le Secret du client requis par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Access token	Renseignez le jeton d'accès requis par Box afin d'accéder au compte Box et le faire fonctionner. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Refresh Token	Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d'accès. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port dans les champs correspondants qui s'affichent.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé en standalone en tant que sous-job afin de créer la connexion à Box utilisée. Dans un Job, il est souvent relié aux autres composants Box à l'aide de liens **Trigger** tel que le lien **OnSubjobOk**.	
Limitation	n/a	

Scénario associé

Pour un scénario associé, consultez *Scénario : Charger et télécharger des fichiers depuis Box*.

tBoxCopy

Propriétés du tBoxCopy

Famille de composant	Cloud/Box	
Fonction	Ce composant se connecte à un compte Box et copie ou déplace un fichier ou un dossier donné.	
Objectif	Le composant **tBoxCopy** copie ou déplace un dossier ou un fichier donné au sein de Box.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Connection/Client Key	Saisissez la clé client requise par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Connection/Client Secret	Saisissez le Secret du client requis par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Connection/Access Token	Renseignez le jeton d'accès requis par Box afin d'accéder au compte Box et le faire fonctionner. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Connection/Refresh Token	Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d'accès. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Connection/Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port dans les champs correspondants qui s'affichent.
	Move Directory	Cochez cette case afin de déplacer un dossier au sein de Box.
	Copy Directory	Cochez cette case afin de copier un dossier au sein de Box.
	File Name	Saisissez le nom du fichier que vous souhaitez copier ainsi que son chemin dans Box.
	Source Directory	Cette option n'est disponible que si vous avez coché la case **Move Directory** ou la case **Copy Directory**. Saisissez le nom du répertoire source dans Box à copier ou à déplacer.
	Destination Directory	Saisissez le nom du dossier cible dans Box dans lequel vous souhaitez copier ou déplacer le fichier ou le dossier spécifié.
	Rename	Cochez cette case afin de renommer le fichier ou le dossier copié. Lorsque vous copiez un fichier, saisissez le nouveau nom du fichier dans le champ **Destination File Name**. Lorsque vous copiez un dossier, saisissez le nouveau nom du dossier dans le champ **New Directory Name**.
	Remove Source File	Cochez cette case afin de supprimer le fichier source pendant l'action de copie.

	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Notez que le schéma de ce composant est en lecture seule. Il a quatre colonnes nommées **destinationFilePath**, **destinationFileName**, **sourceDirectory** et **destinationDirectory**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **DESTINATION_FILENAME** : nom du fichier cible. Cette variable est une variable **After** et retourne une chaîne de caractères. **DESTINATION_FILEPATH** : chemin d'accès au fichier cible. Cette variable est une variable **After** et retourne une chaîne de caractères. **SOURCE_DIRECTORY** : répertoire source. Cette variable est une variable **After** et retourne une chaîne de caractères. **DESTINATION_DIRECTORY** : répertoire cible. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en standalone dans un sous-job afin de copier ou déplacer des données au sein de Box.	
Limitation	n/a	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tBoxDelete

Propriétés du tBoxDelete

Famille de composant	Cloud/Box	
Fonction	Ce composant se connecte à un compte Box et supprime un fichier ou un dossier défini.	
Objectif	Le composant **tBoxDelete** supprime un fichier ou dossier depuis Box.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Connection/Client Key	Saisissez la clé client requise par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Connection/Client Secret	Saisissez le Secret du client requis par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Connection/Access Token	Renseignez le jeton d'accès requis par Box afin d'accéder au compte Box et le faire fonctionner. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Connection/Refresh Token	Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d'accès. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Connection/Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port dans les champs correspondants qui s'affichent.
	Path	Saisissez le chemin sur Box pointant vers le dossier ou le fichier que vous souhaitez supprimer.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Notez que le schéma de ce composant est en lecture seule. Il contient une seule colonne nommée **filepath**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **REMOVED_PATH** : chemin d'accès au dossier ou fichier supprimé sur Box. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en standalone dans un sous-job afin de supprimer des données depuis Box.
Limitation	n/a

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tBoxGet

Propriétés du tBoxGet

Famille de composant	Cloud/Box	
Fonction	Ce composant se connecte à un compte Box donné et télécharge des fichiers vers un répertoire local défini.	
Objectif	Le composant **tBoxGet** télécharge un fichier depuis un compte Box.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Connection/Client Key	Saisissez la clé client requise par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Connection/Client Secret	Saisissez le Secret du client requis par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Connection/Access Token	Renseignez le jeton d'accès requis par Box afin d'accéder au compte Box et le faire fonctionner. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Connection/Refresh Token	Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d'accès. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Connection/Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port dans les champs correspondants qui s'affichent.
	Path	Saisissez le chemin d'accès à Box pointant vers le fichier à télécharger.
	Save as file	Cochez cette case pour afficher le champ **Save To** et parcourez votre système, ou saisissez le chemin d'accès au répertoire local dans lequel stocker le fichier téléchargé. Si le fichier existe, il est remplacé.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Notez que le schéma de ce composant est en lecture seule. Il contient deux colonnes nommées **fileName** et **content**. Le champ **Schema** n'est disponible que si vous cochez la case **Save as file**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du composant.

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **FILE_NAME** : nom du fichier traité. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **INPUT_STREAM** : le contenu du fichier récupéré. Cette variable est une variable **Flow** et retourne une classe InputStream. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé seul ou avec d'autres composants via un lien **Iterate** ou un lien **Trigger** tel que **OnSubjobOk**.
Limitation	n/a

Scénario associé

Pour un scénario associé, consultez *Scénario : Charger et télécharger des fichiers depuis Box*.

tBoxList

Propriétés du tBoxList

Famille de composant	Cloud/Box	
Fonction	Ce composant lit un (des) fichier(s) sur Box dans le répertoire défini et liste les métadonnées et les contenus de ce (ces) fichier(s).	
Objectif	Le composant **tBoxList** liste des fichiers stockés dans un répertoire défini sur Box.	
	Connection/Client Key	Saisissez la clé client requise par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Connection/Client Secret	Saisissez le Secret du client requis par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Connection/Access Token	Renseignez le jeton d'accès requis par Box afin d'accéder au compte Box et le faire fonctionner. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Connection/Refresh Token	Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d'accès. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Connection/Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port dans les champs correspondants qui s'affichent.
	Path	Saisissez le chemin pointant vers le dossier contenant les fichiers à lister ou saisissez le chemin vers le fichier que vous souhaitez lire.
	List type	Sélectionnez le type de données que vous souhaitez lister dans le chemin défini : **Files**, **Folders** ou **Both**.
	Include subdirectories	Cochez cette case afin de lister des fichiers contenus dans les sous-dossiers en plus de ceux contenus dans le dossier défini dans le champ **Path**.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Notez que le schéma de ce composant est en lecture seule. Il contient six colonnes nommées **name**, **path**, **lastModified**, **size**, **id** et **type**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NAME** : le nom du fichier distant traité. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	FILE_PATH : chemin d'accès au dossier ou fichier à traiter sur Box. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **FILE_DIRECTORY** : répertoire du dossier ou fichier à traiter sur Box. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **LAST_MODIFIED** : date et heure de la dernière modification apportée au fichier traité. Cette variable est une variable **Flow** et retourne un long. **SIZE** : le volume du fichier traité. Cette variable est une variable **Flow** et retourne un long. **ID** : ID du dossier ou fichier à traiter sur Box. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **TYPE** : type des objets à traiter sur Box, fichier ou dossier. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en standalone.
Limitation	n/a

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tBoxPut

Propriétés du tBoxPut

Famille de composant	Cloud/Box	
Fonction	Ce composant charge des données sur Box à partir d'un fichier local ou à partir d'un flux de données.	
Objectif	Le composant **tBoxPut** charge des fichiers sur un compte Box.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Connection/Client Key	Saisissez la clé client requise par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Connection/Client Secret	Saisissez le Secret du client requis par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.
	Connection/Access Token	Renseignez le jeton d'accès requis par Box afin d'accéder au compte Box et le faire fonctionner. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Connection/Refresh Token	Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d'accès. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.
	Connection/Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port dans les champs correspondants qui s'affichent.
	Remote Path	Saisissez le chemin pointant vers le fichier dans lequel vous souhaitez écrire des données. Si le fichier n'existe pas, il est créé à la volée.
	Replace if Existing	Cochez cette case afin de remplacer le fichier existant par le fichier chargé.
	Upload mode	Sélectionnez le mode de chargement utilisé : • **Upload incoming content as file** : Sélectionnez ce bouton radio afin de lire les données directement depuis le flux d'entrée du composant précédent. Ces données sont ensuite écrites dans le fichier spécifié dans le champ **Remote Path**. • **Upload local file** : Sélectionnez ce bouton radio afin de charger un fichier local sur Box. Dans le champ **File** qui s'affiche, saisissez le chemin ou parcourez votre système jusqu'à ce fichier. • **Expose as OutputStream** : Cochez cette case afin d'exposer le flux de sortie de ce composant. Ce flux de sortie peut être utilisé par d'autres composants afin d'écrire des données dans le fichier. Par exemple, vous pouvez utiliser la fontionnalité **Use output stream** du composant **tFileOutputDelimited** afin d'alimenter le flux de sortie d'un **tBoxPut** exposé. Pour plus d'informations, consultez *tFileOutputDelimited*.

	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Notez que le schéma de ce composant est en lecture seule. Il contient une seule colonne nommée **content** et reçoit des données uniquement depuis la colonne *content* du schéma d'entrée. Cela signifie que vous devez utiliser une colonne *content* dans le flux de données d'entrée afin de transporter les données à charger. Ce type de colonne est généralement fournit par le composant **tFileInputRaw**. Pour plus d'informations, consultez *tFileInputRaw*. Le champ **Schema** n'est disponible que si vous avez sélectionné le mode de chargement **Expose as OutputStream** ou **Upload local file**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé soit en standalone dans un sous-job, afin de charger un fichier local directement sur Box, ou en tant que composant de fin du flux d'un Job afin de charger des données traitées par ce flux.	
Limitation	n/a	

Scénario : Charger et télécharger des fichiers depuis Box

Ce scénario décrit un Job de trois composants qui charge un fichier sur Box puis télécharge un fichier depuis Box sur votre système de fichiers local.

Avant de reproduire ce scénario, vous devez créer un compte sur https://developers.box.com/ puis créer une application Box sous le compte Box utilisé. Pour plus d'informations sur l'application Box, consultez https://app.box.com/developers/services/edit/. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez l'exemple d'application disponible sur https://github.com/box/box-java-sdk-v2/wiki/HelloWorld.

Relier les composants

1. Dans la perspective **Integration** du Studio, créez un Job vierge depuis le nœud **Job Designs** du **Repository**.

   Pour plus d'informations sur la création d'un Job, consultez le *Guide utilisateur du Studio Talend*.

2. Dans l'espace de modélisation graphique, saisissez le nom du composant à utiliser et sélectionnez ce composant dans la liste qui s'ouvre. Dans ce scénario, les composants sont le **tBoxConnection**, le **tBoxPut** et le **tBoxGet**.

3. Reliez le **tBoxConnection** au **tBoxPut** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

4. Reliez le **tBoxPut** au **tBoxGet** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Configurez les composants

1. Double-cliquez sur le **tBoxConnection** afin d'ouvir sa vue **Component**.

2. Renseignez la clé client, le Secret du client, le jeton d'accès et le jeton de rafraîchissement entre guillemets doubles dans les champs correspondants afin d'accéder au compte Box.

3. Double-cliquez sur le **tBoxPut** afin d'ouvrir sa vue **Component**.

4. Cochez la case **Use Existing Connection** afin de réutiliser la connexion créée par le composant **tBoxConnection**.

   Dans le champ **Remote Path**, saisissez le chemin vers lequel vous souhaitez charger le fichier.

   Dans la zone **Upload mode**, sélectionnez **Upload Local File** Dans le champ **File**, saisissez le chemin ou parcourez votre système vers le fichier que vous souhaitez charger.

5. Double-cliquez sur le **tBoxGet** afin d'ouvrir sa vue **Component**.

6. Cochez la case **Use Existing Connection** afin de réutiliser la connexion créée par le **tBoxConnection**.

   Dans le champ **Path**, saisissez le chemin du fichier que vous souhaitez télécharger.

   Cochez la case **Save As File**. Dans le champ **Save To**, saisissez le chemin vers lequel vous souhaitez sauvegarder le fichier dans votre système local.

7. Sauvegardez le Job.

Exécuter le Job

Exécutez le Job en appuyant sur **F6** ou en cliquant sur le bouton **Run** dans l'onglet **Run**.

Le fichier local, *hello.txt* dans cet exemple, est chargé sur votre Box.

Le fichier *box.txt* est téléchargé depuis Box vers votre système de fichiers local.

« New Volum... ▸ data ▸
Search data
Organize
Open
Print
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Name
Date modified
Type
input
9/29/2014 12:04 PM
File folder
output
3/11/2014 6:01 PM
File folder
temp
1/13/2014 2:27 PM
File folder
box.txt
9/29/2014 2:18 PM
Text Document

tCloudStart

Propriétés du tCloudStart

Famille de composant	Cloud	
Fonction	Le composant **tCloudStart** accède au fournisseur Cloud à utiliser (Amazon EC2) et lance des instances, qui sont des serveurs virtuels dans ce Cloud. Si une instance à lancer n'existe pas, le **tCloudStart** la crée.	
Objectif	Ce composant démarre des instances sur Amazon EC2 (Amazon Elastic Compute Cloud).	
Basic settings	*Access key* et *Secret key*	Saisissez ou collez la clé d'accès et la clé secrète requises par Amazon pour authentifier vos requêtes dans ses services Web. Ces informations d'accès sont générées dans l'onglet Security Credential de la page de votre compte Amazon. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Cloud provider	Sélectionnez le fournisseur Cloud à utiliser.
	Image	Saisissez le nom de l'image de la machine Amazon (Amazon Machine Image) à utiliser pour lancer une instance. L'AMI définit la configuration de base de cette instance.
	Region et *Zone*	Saisissez la région et la zone à utiliser comme emplacement géographique où lancer une instance. La syntaxe utilisée pour exprimer un lieu est prédéfinie par Amazon. Par exemple, *us-east-1* représente la région *US East (Northern Virginia)* et *us-east-1a* représente l'une es zones de disponibilité dans cette région. Pour plus d'informations à propos des régions disponibles pour Amazon, consultez la documentation Amazon concernant les régions et les endpoints ainsi que la FAQ Amazon des régions et des zones de disponibilité.
	Instance name	Saisissez le nom de l'instance à lancer. Par exemple, vous pouvez saisir *Talend*. Notez que toute lettre en majuscule sera convertie en minuscule.
	Instance count	Saisissez le nombre d'instances à lancer. Au moment de l'exécution, le nom spécifié dans le champ **Instance name**, par exemple *Talend*, est utilisé comme première partie du nom de chaque instance. Des lettres et des chiffres sont ajoutés au hasard afin de compléter chaque nom.
	Instance type	Sélectionnez le type d'instance(s) à lancer. Chaque type est prédéfini par Amazon et définit la performance de chaque instance à lancer. Cette liste déroulante présente le nom de l'API pour chaque type d'instance. Pour plus d'informations, consultez la documentation Amazon concernant les types d'instance.
	Proceed with a Key pair	Cochez cette case pour utiliser la paire de clés Amazon (Key Pair) afin de vous connecter à Amazon EC2. Une fois la case cochée, une liste déroulante apparaît. Vous pouvez sélectionner : • **Use an existing Key Pair** pour saisir le nom de cette paire de clés dans le champ à côté de la liste déroulante. Si nécessaire,

		Amazon vous demande de trouver et d'utiliser cette paire de clés lors de l'exécution. • **Create a Key Pair** pour saisir le nom de la nouvelle paire de clés dans le champ à côté de la liste déroulante et définir l'emplacement où stocker cette paire de clés, dans la vue **Advanced settings**.
	Security group	Ajoutez des lignes dans cette table et saisissez les noms des groupes de sécurité auxquels vous devez assigner l'instance ou les instances à lancer. Les groupes de sécurité configurés dans cette table doivent exister dans votre Amazon EC2. Un groupe de sécurité applique des règles spécifiques sur le trafic entrant dans les instances assignées au groupe, comme les ports à utiliser. Pour plus d'informations concernant les groupes de sécurité, consultez la documentation Amazon concernant les groupes de sécurité.
Advanced settings	*Key Pair folder*	Parcourez votre système, ou saisissez le chemin d'accès au dossier à utiliser pour stocker la paire de clés créée. Ce champ apparaît lorsque vous sélectionnez **Creating a Key Pair** dans l'onglet **Basic settings**.
	Volumes	Ajoutez des lignes et définissez le(s) volume(s) à créer pour les instances à lancer, en plus des volumes prédéfinis et alloués par votre Amazon EC2. Les paramètres à configurer dans cette table ont la même valeur que les paramètres utilisés Amazon pour décrire un volume. Si vous devez supprimer automatiquement un volume supplémentaire après la fin de son instance relative, cochez la case dans la colonne **Delete on termination**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
Global Variables	**NODE_GROUP** : nom de l'instance. Cette variable est une variable **After** et retourne une chaîne de caractères. **NODES** : instances en cours d'exécution. Cette variable est une variable **After** et retourne un object. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone afin de lancer une instance de Amazon EC2. Vous pouvez utiliser ce composant pour démarrer l'instance sur laquelle déployer des Jobs.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tCloudStop

Propriétés du tCloudStop

Famille de composant	Cloud	
Fonction	Le composant **tCloudStart** accède au fournisseur Cloud à utiliser (Amazon EC2) et suspend, reprend ou termine une ou des instance(s) données(s).	
Objectif	Ce composant vous permet de modifier le statut d'une instance lancée sur Amazon EC2 (Amazon Elastic Compute Cloud).	
Basic settings	*Access key* et *Secret key*	Saisissez ou collez la clé d'accès et la clé secrète requises par Amazon pour authentifier vos requêtes dans ses services Web. Ces informations d'accès sont générées dans l'onglet Security Credential de la page de votre compte Amazon. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Cloud provider	Sélectionnez le fournisseur Cloud à utiliser.
	Action	Sélectionnez l'action que doit effectuer le **tCloudStop** afin de modifier le statut d'une instance donnée. L'action peut être : • **Suspend** • **Resume** • **Terminate** Notez que si vous terminez une instance, cette instance est supprimée, alors que vous pouvez toujours reprendre une instance suspendue.
	Predicate	Sélectionnez l'instance ou les instances dont vous souhaitez modifier le statut. Les options sont les suivantes : • **Running instances** : le statut de toutes les instances en cours est modifié. • **Instances in a specific group** : le statut des instances d'un groupe d'instances spécifique est modifié. Vous devez saisir le nom de ce groupe dans le champ **Group name**. • **Running instances in a specific group** : le statut des instances en cours d'un groupe d'instances spécifiques est modifié. Vous devez saisir le nom de ce groupe dans le champ **Group name**. • **Instance with predefined id** : le statut d'une instance donnée est modifié. Vous devez saisir l'ID de cette instance dans le champ **Id**. Vous pouvez trouver cet ID dans votre Amazon EC2. Un groupe d'instances se compose d'instances utilisant le même nom d'instance, défini dans le champ **Instance name** du composant **tCloudStart**.
	Group name	Saisissez le nom du groupe dans lequel vous souhaitez modifier le statut des instances données. Ce champ apparaît lorsque vous sélectionnez **Instances in a specific group** ou **Running instances in a specific group** dans la liste **Predicate**.

	Id	Saisissez l'ID de l'instance dont vous devez changer le statut. Le champ apparaît lorsque vous sélectionnez **Instance with predefined id** dans la liste **Predicate**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau des composants.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant fonctionne en standalone pour changer le statut d'instances données d'Amazon EC2. Vous pouvez utiliser ce composant afin de suspendre, reprendre ou terminer une ou des instance(s) où sont déployés des Jobs. Ce composant est généralement utilisé avec le **tCloudStart** afin de changer le statut des instances lancées par celui-ci.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tDropboxConnection

Propriétés du tDropboxConnection

Famille de composant	Cloud/Dropbox	
Fonction	Ce composant crée une connexion vers un compte Dropbox donné.	
Objectif	Le composant **tDropboxConnection** crée une connexion Dropbox que les autres composants Dropbox peuvent réutiliser.	
Basic settings	*Access token*	Renseignez le jeton d'accès requis par le compte Dropbox auquel vous souhaitez vous connecter. Ce jeton d'accès permet au Studio d'appeler l'API Dropbox pour ce compte Dropbox. Notez qu'une application Dropbox doit avoir été créée au préalable sous ce compte avant que le jeton d'accès puisse être généré. Pour plus d'informations concernant les jetons d'accès Dropbox, consultez https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account (en anglais).
	Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port de ce proxy dans les champs correspondants qui s'affichent.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau des composants.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé en standalone en tant que sous-job pour créer une connexion Dropbox. Dans un Job, ce composant est généralement connecté aux autres composants Dropbox à l'aide de liens de type **Trigger**, tel que le lien **On Subjob Ok**.	
Limitation	n/a	

Scénario associé

Pour un scénario associé, consultez *Scénario : Télécharger des fichiers dans Dropbox*

tDropboxDelete

Propriétés du tDropboxDelete

Famille de composant	Cloud/Dropbox	
Fonction	Ce composant se connecte à un compte Dropbox donné et supprime un fichier ou un dossier spécifié.	
Objectif	Le composant **tDropboxDelete** supprime un dossier ou un fichier donné de Dropbox.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access token	Renseignez le jeton d'accès requis par le compte Dropbox auquel vous souhaitez vous connecter. Ce jeton d'accès permet au Studio d'appeler l'API Dropbox pour ce compte Dropbox. Notez qu'une application Dropbox doit avoir été créée au préalable sous ce compte avant que le jeton d'accès puisse être généré. Pour plus d'informations concernant les jetons d'accès Dropbox, consultez https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account (en anglais).
	Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port de ce proxy dans les champs correspondants qui s'affichent.
	Path	Renseignez le chemin Dropbox d'accès au dossier ou fichier que vous souhaitez supprimer. Notez que le nom du chemin d'accès doit commencer par une barre oblique (/). Il s'agit du dossier racine de l'application Dropbox pour laquelle vous utilisez le jeton d'accès.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau des composants.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé en standalone dans un sous-job pour supprimer des données de Dropbox.	
Limitation	n/a	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tDropboxGet

Propriétés du tDropboxGet

Famille de composant	Cloud/Dropbox	
Fonction	Ce composant se connecte à un compte Dropbox et télécharge des fichiers dans un répertoire local spécifié.	
Objectif	Le composant **tDropboxGet** télécharge un fichier sélectionné d'un compte Dropbox.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access token	Renseignez le jeton d'accès requis par le compte Dropbox auquel vous souhaitez vous connecter. Ce jeton d'accès permet au Studio d'appeler l'API Dropbox pour ce compte Dropbox. Notez qu'une application Dropbox doit avoir été créée au préalable sous ce compte avant que le jeton d'accès puisse être généré. Pour plus d'informations concernant les jetons d'accès Dropbox, consultez https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account (en anglais).
	Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port de ce proxy dans les champs correspondants qui s'affichent.
	Path	Renseignez le chemin Dropbox d'accès au fichier que vous souhaitez télécharger. Notez que le nom du chemin d'accès doit commencer par une barre oblique (/). Il s'agit du dossier racine de l'application Dropbox pour laquelle vous utilisez le jeton d'accès.
	Save as file	Cochez cette case pour afficher le champ **File** et parcourez votre système ou saisissez le chemin d'accès au répertoire local dans lequel vous souhaitez que soit stocké le fichier téléchargé. Le fichier est remplacé s'il existe déjà.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau des composants.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **FILE_NAME** : nom du fichier traité. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **OUTPUT_PATH** : le répertoire dans lequel les fichiers de sortie sont stockés. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone ou avec d'autres composants à l'aide d'un lien de type **Iterate** ou d'un lien de type **Trigger** tel que **OnSubjobOK**.	

Limitation	n/a

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tDropboxList

Propriétés du tDropboxList

Famille de composant	Cloud/Dropbox	
Fonction	Ce composant lit le ou les fichier(s) dans Dropbox contenu(s) dans le répertoire que vous spécifiez et liste les métadonnées ainsi que le contenu de ce ou ces fichier(s).	
Objectif	Le composant **tDropboxList** liste les fichiers stockés dans un répertoire spécifié dans Dropbox.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access token	Renseignez le jeton d'accès requis par le compte Dropbox auquel vous souhaitez vous connecter. Ce jeton d'accès permet au Studio d'appeler l'API Dropbox pour ce compte Dropbox. Notez qu'une application Dropbox doit avoir été créée au préalable sous ce compte avant que le jeton d'accès puisse être généré. Pour plus d'informations concernant les jetons d'accès Dropbox, consultez https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account (en anglais).
	Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port de ce proxy dans les champs correspondants qui s'affichent.
	Path	Renseignez le chemin d'accès vers le dossier dont vous souhaitez lister les fichiers, ou renseignez le chemin d'accès vers le fichier exact que vous souhaitez lire. Notez que le nom du chemin d'accès doit commencer par une barre oblique (/). Il s'agit du dossier racine de l'application Dropbox pour laquelle vous utilisez le jeton d'accès.
	List type	Sélectionnez, dans le chemin d'accès spécifié, le type de données que vous souhaitez lister.
	Include subdirectories	Cochez cette case pour lister les fichiers de n'importe quel sous-dossier existant en plus des fichiers dans le répertoire défini dans le champ **Path**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau des composants.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NAME** : le nom du fichier distant traité. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **PATH** : le chemin vers le dossier ou le fichier traité dans Dropbox. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **LAST_MODIFIED** : date et heure de la dernière modification apportée au fichier traité. Cette variable est une variable **Flow** et retourne un long. **SIZE** : le volume du fichier traité. Cette variable est une variable **Flow** et retourne un long. **IS_FILE** : le résultat booléen de la liste de fichiers. Cette variable est une variable **Flow** et retourne un booléen. Le résultat *Yes* indique que les données listées sont de type File. Sinon, le type est Folder.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en standalone.
Limitation	n/a

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tDropboxPut

Propriétés du tDropboxPut

Famille de composant	Cloud/Dropbox	
Fonction	Ce composant télécharge des données dans Dropbox à partir d'un fichier local ou d'un flux de données.	
Objectif	Le composant **tDropboxPut** télécharge des fichiers dans un compte Dropbox.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access token	Renseignez le jeton d'accès requis par le compte Dropbox auquel vous souhaitez vous connecter. Ce jeton d'accès permet au Studio d'appeler l'API Dropbox pour ce compte Dropbox. Notez qu'une application Dropbox doit avoir été créée au préalable sous ce compte avant que le jeton d'accès puisse être généré. Pour plus d'informations concernant les jetons d'accès Dropbox, consultez https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account (en anglais).
	Use HTTP proxy	Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port de ce proxy dans les champs correspondants qui s'affichent.
	Path (File only)	Renseignez le chemin d'accès vers le fichier dans lequel vous souhaitez écrire du contenu. Ce fichier est créé s'il n'existe pas. Notez que le nom du chemin d'accès doit commencer par une barre oblique (/). Il s'agit du dossier racine de l'application Dropbox pour laquelle vous utilisez le jeton d'accès.
	Upload mode	Sélectionnez le mode de téléchargement à utiliser : • **Rename if existing** : le fichier téléchargé dans Dropbox est automatiquement renommé. Par exemple, un fichier nommé *test.txt* peut être renommé en *test (1)*.txt. • **Replace if existing** : le fichier téléchargé dans Dropbox remplace le fichier déjà existant. • **Update specified revision** : le fichier que vous téléchargez dans Dropbox est utilisé pour mettre à jour une révision spécifique de ce fichier. Si la révision que vous spécifiez est la dernière révision, alors le fichier existant dans Dropbox est remplacé. S'il s'agit une révision antérieure, le fichier que vous téléchargez est renommé pour indiquer qu'il y a un conflit. Si la révision n'existe pas, une erreur est retournée.
	Upload incoming content as file	Sélectionnez ce bouton radio pour lire les données directement du flux d'entrée du composant précédent et écrire les données dans le fichier spécifié dans le champ **Path**.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Notez que le schéma de ce composant est en lecture seule et est composé d'une seule colonne appelée **content**. Il reçoit des données de la colonne *content* de son schéma d'entrée uniquement. Cela signifie que vous devez utiliser une colonne *content* dans le flux de données d'entrée pour transporter les données à télécharger. Ce type de colonne

		est généralement fourni par le composant **tFileInputRaw**. Pour plus d'informations, consultez *tFileInputRaw*. Le champ **Schema** n'est pas disponible lorsque vous sélectionnez le bouton radio **Expose as OutputStream** ou **Upload local file**.
	Upload local file	Sélectionnez ce bouton radio pour télécharger dans Dropbox un fichier stocké localement. Dans le champ **File** qui s'affiche, vous devez saisir le chemin d'accès ou parcourir votre système jusqu'au fichier.
	Expose as OutputStream	Cochez cette case pour exposer le flux de sortie de ce composant en tant que variable appelée **OUTPUTSTREAM**, afin que les autres composants puissent réutiliser cette variable pour écrire le contenu à télécharger dans le flux de sortie exposé. Par exemple, vous pouvez utiliser la fonctionnalité **Use output stream** du composant **tFileOutputDelimited** pour alimenter le flux de sortie exposé d'un **tDropboxPut** donné. Pour plus d'informations, consultez *tFileOutputDelimited*.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau des composants.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **OUTPUTSTREAM** : le contenu du fichier mis en ligne. Cette variable est une variable **Flow** et retourne un object. Notez que cette variable **OUTPUTSTREAM** est disponible uniquement lorsque la case **Expose as OutputStream** est cochée. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est soit utilisé en standalone dans un sous-job pour télécharger un fichier local directement dans Dropbox, soit utilisé en tant que composant de sortie d'un flux de Job pour télécharger des données manipulées dans ce flux.	
Limitation	n/a	

Scénario : Télécharger des fichiers dans Dropbox

Dans ce scénario, un Job à six composants composé de trois sous-jobs est créé pour écrire des données dans Dropbox à l'aide de différents modes de téléchargement.

Avant de reproduire ce scénario, vous devez créer une application Dropbox sous le compte Dropbox utilisé. Dans ce scénario, l'application Dropbox utilisée est appelée *talenddrop*. Le fichier racine dans lequel les fichiers sont téléchargés est appelé *talenddrop*également. En outre, le jeton d'accès à ce fichier a été généré à partir de la console de l'application fournie par Dropbox.

Pour plus d'informations concernant l'application Dropbox, consultez https://www.dropbox.com/developers/apps/ .

Relier les composants

1. Dans la perspective **Integration** du Studio, créez un Job vide à partir du nœud **Job Designs** dans la vue **Repository**.

   Pour plus d'informations concernant la création d'un Job, consultez le *Guide utilisateur du Studio Talend*.

2. Dans l'espace de modélisation, saisissez le nom du composant à utiliser et sélectionnez ce composant dans la liste qui apparaît. Dans ce scénario, les composants utilisés sont le **tDropboxConnection**, le **tFixedFlowInput**, le **tFileOutputDelimited**, le **tFileInputRaw** et deux composants **tDropboxPut**.

   Dans ce scénario, le composant **tFixedFlowInput** génère des données à télécharger dans Dropbox. Dans des cas réels, vous pouvez utiliser d'autres composants tels que le **tMysqlInput** ou le **tMap** à la place du **tFixedFlowInput**, afin de concevoir un processus plus élaboré pour préparer le traitement de vos données.

3. Reliez le **tFixedFlowInput** au **tFileOutputDelimited** à l'aide d'un lien de type **Row > Main**.

4. Faites de même pour relier le **tFileOutputDelimited** à l'un des deux composants **tDropboxPut** et reliez le **tFileInputRaw** à l'autre composant **tDropboxPut**.

5. Reliez le **tDropboxConnection** au **tFixedFlowInput** à l'aide d'un lien de type **Trigger > OnSubjobOk**, puis reliez le **tFixedFlowInput** au **tFileInputRaw** à l'aide d'un lien de même type.

Se connecter à Dropbox

1. Double-cliquez sur le **tDropboxConnection** pour ouvrir sa vue **Component**.

2. Dans le champ **Access token**, collez le jeton que vous avez généré via la console de l'application de Dropbox afin d'accéder au dossier de l'application Dropbox à utiliser.

Générer le flux de sortie

Définir les données d'entrée

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Component**.

Dans ce scénario, seules trois lignes de données sont créées pour indiquer trois pays et leurs indicatifs téléphoniques.

```
33;France
86;China
81;Japan
```

2. Cliquez sur le bouton **[...]** à côté de **Edit schema** pour ouvrir l'éditeur de schéma.

3. Cliquez deux fois sur le bouton **[+]** pour ajouter deux lignes et, dans la colonne **Column**, renommez-les *code* et *country*.

4. Cliquez sur **OK** pour valider ces changements et accepter la propagation demandée par la boîte de dialogue qui s'ouvre.

5. Dans la zone **Mode**, sélectionnez le bouton radio **Use Inline Table**. Les colonnes *code* et *country* sont automatiquement créées dans cette table.

6. Saisissez dans cette table les données mentionnées ci-dessus.

Définir le flux de sortie

1. Double-cliquez sur le **tFileOutputDelimited** pour ouvrir sa vue **Component**.

2. Cochez la case **Use output stream** pour écrire les données dans un flux de sortie donné.

3. Dans le champ **Output stream**, renseignez le code pour définir le flux de sortie dans lequel vous souhaitez écrire les données. Dans ce scénario, il s'agit du flux de sortie du composant **tDropboxPut_1** relié au composant actuel. Le code utilisé pour écrire les données se présente de la manière suivante :

```
((java.io.OutputStream)globalMap.get("tDropboxPut_1_OUTPUTSTREAM"))
```

   Notez que dans cet exemple de code, le composant **tDropboxPut** a le chiffre *1* comme affixe. Il correspond à son ID de composant automatiquement attribué dans ce Job. Si le composant **tDropboxPut** que vous utilisez a un ID différent, vous devez adapter le code en fonction de cet ID.

4. Cliquez sur **Edit schema** pour vérifier que le schéma de ce composant est identique à celui du composant **tFixedFlowInput** qui précède. Si ce n'est pas le cas, cliquez sur le bouton **Sync columns** pour rendre les deux schémas identiques.

Exposer le flux de sortie du tDropboxPut

1. Double-cliquez sur le composant **tDropboxPut** relié au **tFileOutputDelimited** pour ouvrir sa vue **Component**.

2. Cochez la case **Use existing connection** pour réutiliser la connexion créée par le **tDropboxConnection**.

3. Dans le champ **Path**, saisissez le chemin d'accès au fichier dans lequel vous souhaitez écrire les données en le faisant précéder d'une barre oblique (/) . Par exemple, saisissez */calling_code.csv*.

4. Dans la zone **Upload mode**, sélectionnez le bouton radio **Rename if Existing**.

5. Sélectionnez le bouton radio **Expose As OutputStream** pour exposer le flux de sortie de ce composant afin que l'autre composant, **tFileOutputDelimited** dans ce scénario, puisse écrire des données dans le flux.

Définir les données du média à télécharger

1. Double-cliquez sur le **tFileInputRaw** pour ouvrir sa vue **Component**.

Ce composant est utilisé pour lire une image appelée *esb_architecture.png* dans le flux de données. Dans des cas réels, ce fichier peut avoir de nombreux autres formats tels que PDF, XLS, PPT ou MP3.

2. Dans le champ **Filename**, saisissez le chemin d'accès ou parcourez votre système jusqu'au fichier que vous souhaitez télécharger.

3. Dans la zone **Mode**, sélectionnez le bouton radio **Read the file as a bytes array**.

Mettre à jour le contenu entrant

1. Double-cliquez sur le composant **tDropboxPut** relié au **tFileInputRaw** pour ouvrir sa vue **Component**.

2. Cochez la case **Use existing connection** pour réutiliser la connexion créée par le **tDropboxConnection**.
3. Dans le champ **Path**, saisissez le chemin d'accès vers le fichier dans lequel vous souhaitez écrire les données en le faisant précéder d'une barre oblique (/) . Par exemple, saisissez */architecture.png*.
4. Dans la zone **Upload mode**, sélectionnez **Rename if existing**.
5. Sélectionnez le bouton **Upload incoming content as file**. Cela fait apparaître le bouton **Edit schema** pour vous permettre de voir le schéma en lecture seule de ce composant.

Exécuter le Job

Appuyez sur **F6** pour exécuter ce Job.

Lorsque vous avez terminé, vérifiez les fichiers téléchargés dans le dossier spécifié de votre application Dropbox. Dans ce scénario, il s'agit du dossier *talenddrop*.

tGoogleDriveConnection

Propriétés du tGoogleDriveConnection

Famille de composant	Cloud / Google Drive	
Fonction	Le composant **tGoogleDriveConnection** ouvre une connexion afin d'accéder à un compte Google Drive.	
Objectif	Ce composant vous permet d'ouvrir une connexion à Google Drive pouvant être réutilisée par d'autres composants Google Drive.	
Basic settings	*Application Name*	Saisissez le nom de l'application requis par Google Drive pour accéder à son APIs.
	Access Token	Saisissez le jeton d'accès requis par Google Drive pour accéder à un compte Google Drive. Pour plus d'informations concernant la récupération du jeton d'accès, consultez *Obtenir un jeton d'accès à Google Drive*.
	Use HTTP Proxy	Cochez cette case si vous travaillez derrière un proxy HTTP.
	Proxy Host	Saisissez l'adresse IP du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Proxy Port	Saisissez le numéro du port du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Use SSL	Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive.
	Algorithm	Saisissez le nom de l'algorithme de cryptographie SSL. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	TrustStore File	Parcourez votre système ou saisissez le chemin d'accès au fichier du certificat TrustStore contenant la liste des certificats acceptés par l'utilisateur. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Password	Saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Google Drive. Dans un Job, il est généralement utilisé pour ouvrir une connexion à Google Drive pouvant être réutilisée par d'autres composants.	
Limitation	n/a	

Obtenir un jeton d'accès à Google Drive

Pour utiliser les API Google Drive, vous devez obtenir le jeton d'accès en complétant les étapes suivantes dans Google Chrome ayant l'extension REST Console installée :

1. Allez à la page Google Developers Console et sélectionnez un projet existant ou créez-en un nouveau. Dans cet exemple, créez un nouveau projet *TalendProject*.

2. Sélectionnez le nouveau projet, cliquez sur **APIs** sous **APIs & auth** et activez les API **Drive API** et **Drive SDK** en changeant leur statut à **ON**.

3. Cliquez sur **Consent screen** sous **APIs & auth** et créez un nouveau projet. Dans cet exemple, nommez-le *TalendProduct*.

4. Cliquez sur **Credentials** sous **APIs & auth**, créez un nouvel ID client et sélectionnez, dans la zone **APPLICATION TYPE** le type **Installed application**. Dans la zone **INSTALLED APPLICATION TYPE**, sélectionnez **Other**.

Les **CLIENT ID**, **CLIENT SECRET** et **REDIRECT URIS** sont utilisés pour obtenir le jeton d'accès.

5. Générez le code d'authentification en exécutant une requête HTTP GET à l'aide d'un navigateur Web. Pour ce faire, copiez-collez l'URL de la requête *https://accounts.google.com/o/oauth2/auth* avec les paramètres suivants dans la barre d'adresse de votre navigateur, appuyez sur la touche **Entrée** et acceptez la demande d'accès.

   - **client_id** : valeur de **CLIENT ID** dans la vue **Credentials**.
   - **redirect_uri** : valeur de **REDIRECT URIS** dans la vue **Credentials**.
   - **response_type** : valeur *code*.
   - **scope** : permissions que vous demandez à l'utilisateur d'accepter pour votre application. Pour plus d'informations, consultez https://developers.google.com/drive/web/scopes (en anglais).
   - **access_type** : valeur *online*.

Notez que l'URL de la requête et les paramètres de la chaîne de caractères sont concaténés par un ? et les paramètres de la requête sont concaténés avec un &. Une URL d'exemple est affichée ci-dessous, avec des retours à la ligne pour des questions de lisibilité. Vous devez regrouper ces lignes en une seule avant de copier-coller l'URL dans la barre d'adresse de votre navigateur. Les permissions dans cet exemple fournissent un accès total à tous les fichiers de l'utilisateur. Demandez ces permissions uniquement lorsque c'est vraiment nécessaire.

```
https://accounts.google.com/o/oauth2/auth?
client_id=1061335238805-gmctuidvid0lsrdf66nvit0g7j5gnrck.apps.googleusercontent.com&
redirect_uri=urn:ietf:wg:oauth:2.0:oob&
response_type=code&
scope=https://www.googleapis.com/auth/drive&
access_type=online
```

La réponse retournée contient un code d'autorisation à utiliser pour générer le jeton d'accès dans l'étape suivante.

6. Générez le jeton d'accès en exécutant une requête HTTP POST à l'aide de Google REST Console. Pour ce faire, copiez-collez l'URL de la requête *https://www.googleapis.com/oauth2/v3/token* avec les paramètres suivants de la requête dans le champ **Request URI** de la zone **Target**, cochez la case **Content-Type** et configurez sa valeur à *application/x-www-form-urlencoded*, puis cliquez sur le bouton **POST**.

   - **code** : code d'autorisation généré dans l'étape précédente.
   - **client_id** : valeur de **CLIENT ID** dans la vue **Credentials**.
   - **client_secret** : valeur de **CLIENT SECRET** dans la vue **Credentials**.
   - **redirect_uri** : valeur de **REDIRECT URIS** dans la vue **Credentials**.
   - **grant_type** : valeur *authorization_code*.

Notez que l'URL de la requête et les paramètres de la chaîne de caractères sont concaténés par un *?* et les paramètres de la requête sont concaténés avec un *&*. Une URL d'exemple est affichée ci-dessous, avec des

retours à la ligne pour des questions de lisibilité. Vous devez regrouper ces lignes en une seule avant de copier-coller l'URL dans le champ **Request URI**.

```
https://www.googleapis.com/oauth2/v3/token?
code=4/fwRIPH2a8XQRCs2JicTs4zWTprKjyJ7KbwkY46fE1O0.sj6y9PmBGe8ZeFc0ZRONyF4NK_eblwI&
client_id=1061335238805-gmctuidvid0lsrdf66nvit0g7j5gnrck.apps.googleusercontent.com&
client_secret=x6Jis-fzAzBfCfrLlokuPopn&
redirect_uri=urn:ietf:wg:oauth:2.0:oob&
grant_type=authorization_code
```

La réponse retournée contient le jeton d'accès. Notez que le jeton d'accès expire au bout de 3600 secondes.

Scénario associé

Pour un scénario associé, consultez *Gestion des fichiers avec Google Drive*.

tGoogleDriveCopy

Propriétés du tGoogleDriveCopy

Famille de composant	Cloud / Google Drive	
Fonction	Le composant **tGoogleDriveCopy** copie ou déplace un fichier ou un dossier spécifié dans Google Drive.	
Objectif	Ce composant vous permet de copier ou déplacer un fichier/dossier dans Google Drive.	
Basic settings	*Use Existing Connection*	Cochez cette case et, dans la liste qui s'affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.
	Application Name	Saisissez le nom de l'application requis par Google Drive pour accéder à son APIs.
	Access Token	Saisissez le jeton d'accès requis par Google Drive pour accéder à un compte Google Drive. Pour plus d'informations concernant la récupération du jeton d'accès, consultez *Obtenir un jeton d'accès à Google Drive*.
	Use HTTP Proxy	Cochez cette case si vous travaillez derrière un proxy HTTP.
	Proxy Host	Saisissez l'adresse IP du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Proxy Port	Saisissez le numéro du port du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Use SSL	Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive.
	Algorithm	Saisissez le nom de l'algorithme de cryptographie SSL. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	TrustStore File	Parcourez votre système ou saisissez le chemin d'accès au fichier du certificat TrustStore contenant la liste des certificats acceptés par l'utilisateur. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Password	Saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	File	Sélectionnez cette option lorsque vous devez copier ou déplacer un fichier.
	Folder	Sélectionnez cette option lorsque vous devez copier un dossier.
	File Name	Saisissez le nom du fichier source à copier ou déplacer. Ce champ est disponible uniquement lorsque l'option **File** est sélectionnée.
	Source Folder Name	Saisissez le nom du dossier source à déplacer. Ce champ est disponible uniquement lorsque l'option **Folder** est sélectionnée.
	Destination Folder Name	Saisissez le nom du dossier de destination dans lequel copier ou déplacer le fichier ou le dossier.
	Rename (set new title)	Cochez cette case afin de renommer le fichier ou dossier copié ou déplacé dans le dossier de destination.
	Destination Name	Saisissez le nom du fichier/dossier après copie ou déplacement dans le dossier de destination.

		Ce champ est disponible uniquement lorsque la case **Rename (set new title)** est cochée.
	Remove Source File	Cochez cette case afin de supprimer le fichier source une fois copié dans le dossier de destination. Ce champ est disponible uniquement lorsque l'option **File** est sélectionnée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton **[...]** à côté de **Edit schema** afin de voir le schéma prédéfini contenant les champs suivants : • **sourceFileID** : ID du fichier/dossier source. • **destinationFileID** : ID du fichier/dossier de destination.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**SOURCE_FILEID** : ID du fichier/dossier source. Cette variable est une variable **After** et retourne une chaîne de caractères. **DESTINATION_FILEID** : ID du fichier/dossier cible. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début, dans un Job ou un sous-job et nécessite un lien de sortie.	
Limitation	n/a	

Scénario associé

Pour un scénario associé, consultez *Gestion des fichiers avec Google Drive*.

tGoogleDriveCreate

Propriétés du tGoogleDriveCreate

Famille de composant	Cloud / Google Drive	
Fonction	Le composant **tGoogleDriveCreate** crée un nouveau dossier dans un dossier parent, dans Google Drive.	
Objectif	Ce composant vous permet de créer un nouveau dossier dans Google Drive.	
Basic settings	*Use Existing Connection*	Cochez cette case et, dans la liste qui s'affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.
	Application Name	Saisissez le nom de l'application requis par Google Drive pour accéder à son APIs.
	Access Token	Saisissez le jeton d'accès requis par Google Drive pour accéder à un compte Google Drive. Pour plus d'informations concernant la récupération du jeton d'accès, consultez *Obtenir un jeton d'accès à Google Drive*.
	Use HTTP Proxy	Cochez cette case si vous travaillez derrière un proxy HTTP.
	Proxy Host	Saisissez l'adresse IP du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Proxy Port	Saisissez le numéro du port du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Use SSL	Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive.
	Algorithm	Saisissez le nom de l'algorithme de cryptographie SSL. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	TrustStore File	Parcourez votre système ou saisissez le chemin d'accès au fichier du certificat TrustStore contenant la liste des certificats acceptés par l'utilisateur. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Password	Saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Parent Folder Name	Saisissez le nom du dossier père dans lequel créer le nouveau dossier.
	New Folder Name	Saisissez le nom du nouveau dossier à créer.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton **[...]** à côté de **Edit schema** afin de voir le schéma prédéfini contenant les champs suivants : • **parentFolderID** : ID du dossier père. • **newFolderID** : ID du nouveau dossier à créer.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**PARENTFOLDERID** : ID du dossier père. Cette variable est une variable **After** et retourne une chaîne de caractères.	

	NEWFOLDERID : ID du nouveau dossier à créer. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en tant que composant de dévit dans un Job ou un sous-job et nécessite un lien de sortie.
Limitation	n/a

Scénario associé

Pour un scénario associé, consultez *Gestion des fichiers avec Google Drive*.

tGoogleDriveDelete

Propriétés du tGoogleDriveDelete

Famille de composant	Cloud / Google Drive	
Fonction	Le composant **tGoogleDriveDelete** supprime un fichier ou un dossier dans Google Drive.	
Objectif	Ce composant vous permet de supprimer un fichier ou un dossier dans Google Drive.	
Basic settings	*Use Existing Connection*	Cochez cette case et, dans la liste qui s'affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.
	Application Name	Saisissez le nom de l'application requis par Google Drive pour accéder à son APIs.
	Access Token	Saisissez le jeton d'accès requis par Google Drive pour accéder à un compte Google Drive. Pour plus d'informations concernant la récupération du jeton d'accès, consultez *Obtenir un jeton d'accès à Google Drive*.
	Use HTTP Proxy	Cochez cette case si vous travaillez derrière un proxy HTTP.
	Proxy Host	Saisissez l'adresse IP du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Proxy Port	Saisissez le numéro du port du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Use SSL	Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive.
	Algorithm	Saisissez le nom de l'algorithme de cryptographie SSL. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	TrustStore File	Parcourez votre système ou saisissez le chemin d'accès au fichier du certificat TrustStore contenant la liste des certificats acceptés par l'utilisateur. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Password	Saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Name of File/Folder To Delete	Saisissez le nom du fichier/dossier à supprimer.
	Use Trash	Cochez cette case afin de déplacer le fichier/dossier à supprimer dans la corbeille.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**FILEID** : ID du champ/dossier à supprimer. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone.
Limitation	n/a

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tGoogleDriveGet

Propriétés du tGoogleDriveGet

Famille de composant	Cloud / Google Drive	
Fonction	Le composant **tGoogleDriveGet** permet de télécharger un fichier stocké dans Google Drive.	
Objectif	Ce composant vous permet de télécharger un fichier dans Google Drive.	
Basic settings	*Use Existing Connection*	Cochez cette case et, dans la liste qui s'affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.
	Application Name	Saisissez le nom de l'application requis par Google Drive pour accéder à son APIs.
	Access Token	Saisissez le jeton d'accès requis par Google Drive pour accéder à un compte Google Drive. Pour plus d'informations concernant la récupération du jeton d'accès, consultez *Obtenir un jeton d'accès à Google Drive*.
	Use HTTP Proxy	Cochez cette case si vous travaillez derrière un proxy HTTP.
	Proxy Host	Saisissez l'adresse IP du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Proxy Port	Saisissez le numéro du port du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Use SSL	Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive.
	Algorithm	Saisissez le nom de l'algorithme de cryptographie SSL. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	TrustStore File	Parcourez votre système ou saisissez le chemin d'accès au fichier du certificat TrustStore contenant la liste des certificats acceptés par l'utilisateur. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Password	Saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Name of File	Saisissez le nom du fichier à télécharger.
	Save As File	Cochez cette case afin de sauvegarder le fichier à télécharger dans un répertoire local.
	Save To	Parcourez votre système ou saisissez le chemin de l'emplacement où sauvegarder le fichier à télécharger. Ce champ est disponible uniquement lorsque la case **Save As File** est cochée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton **[...]** à côté de **Edit schema** afin de voir le schéma prédéfini ayant un champ nommé **content**, décrivant le contenu du fichier à télécharger.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du Job, ainsi qu'au niveau de chaque composant.

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie.
Limitation	n/a

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tGoogleDriveList

Propriétés du tGoogleDriveList

Famille de composant	Cloud / Google Drive	
Fonction	Le composant **tGoogleDriveList** effectue une itération sur le dossier spécifié dans Google Drive.	
Objectif	Ce composant vous permet de lister les fichiers et/ou les sous-dossiers dans un dossier spécifique dans Google Drive.	
Basic settings	*Use Existing Connection*	Cochez cette case et, dans la liste qui s'affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.
	Application Name	Saisissez le nom de l'application requis par Google Drive pour accéder à son APIs.
	Access Token	Saisissez le jeton d'accès requis par Google Drive pour accéder à un compte Google Drive. Pour plus d'informations concernant la récupération du jeton d'accès, consultez *Obtenir un jeton d'accès à Google Drive*.
	Use HTTP Proxy	Cochez cette case si vous travaillez derrière un proxy HTTP.
	Proxy Host	Saisissez l'adresse IP du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Proxy Port	Saisissez le numéro du port du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Use SSL	Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive.
	Algorithm	Saisissez le nom de l'algorithme de cryptographie SSL. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	TrustStore File	Parcourez votre système ou saisissez le chemin d'accès au fichier du certificat TrustStore contenant la liste des certificats acceptés par l'utilisateur. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Password	Saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Folder Name	Saisissez le nom du dossier à itérer.
	FileList Type	Sélectionnez le type de données à lister. • **Files** : les fichiers dans le dossier spécifié. • **Directories** : les sous-dossiers dans le dossier spécifié. • **Both** : Les fichiers et les sous-dossiers dans le dossier spécifié.
	Includes subdirectories	Cochez cette case pour également itérer sur les sous-répertoires.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**ID** : ID du fichier/dossier en cours de traitement. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **TITLE** : titre du fichier/dossier en cours de traitement. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	MIME_TYPE : type MIME du fichier/dossier en cours de traitement. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Pour plus d'informations concernant les types MIME supportés dans Google Drive, consultez Supported MIME Types (en anglais). **LAST_MODIFIED** : dernière date de modification du fichier/dossier en cours de traitement. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **SIZE** : taille du fichier en octets. Cette variable est une variable **Flow** et retourne un long lorsqu'un fichier est en cours de traitement et un null lorsqu'un dossier est en cours de traitement. **SELF_LINK** : lien vers le fichier/dossier en cours de traitement. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **PARENTS** : propriétés du dossier père du fichier/dossier en cours de traitement. Cette variable est une variable **Flow** et retourne une liste. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone ou en tant que composant de début dans un Job ou un sous-job.
Limitation	n/a

Scénario associé

Pour un scénario associé, consultez *Gestion des fichiers avec Google Drive*.

tGoogleDrivePut

Propriétés du tGoogleDrivePut

Famille de composant	Cloud / Google Drive	
Fonction	Le composant **tGoogleDrivePut** charge des données d'un flux de données d'entrée ou d'un fichier local, dans Google Drive.	
Objectif	Ce composant vous permet de charger des données dansGoogle Drive.	
Basic settings	*Use Existing Connection*	Cochez cette case et, dans la liste qui s'affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.
	Application Name	Saisissez le nom de l'application requis par Google Drive pour accéder à son APIs.
	Access Token	Saisissez le jeton d'accès requis par Google Drive pour accéder à un compte Google Drive. Pour plus d'informations concernant la récupération du jeton d'accès, consultez *Obtenir un jeton d'accès à Google Drive*.
	Use HTTP Proxy	Cochez cette case si vous travaillez derrière un proxy HTTP.
	Proxy Host	Saisissez l'adresse IP du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Proxy Port	Saisissez le numéro du port du serveur du proxy HTTP. Ce champ est disponible uniquement lorsque la case **Use HTTP Proxy** est cochée.
	Use SSL	Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive.
	Algorithm	Saisissez le nom de l'algorithme de cryptographie SSL. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	TrustStore File	Parcourez votre système ou saisissez le chemin d'accès au fichier du certificat TrustStore contenant la liste des certificats acceptés par l'utilisateur. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	Password	Saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore. Ce champ est disponible uniquement lorsque la case **Use SSL** est cochée.
	File Title	Saisissez le nom du fichier après chargement.
	Destination Folder Name	Saisissez le nom du dossier dans lequel le fichier sera stocké.
	Replace if Existing	Cochez cette case pour écraser tout fichier existant par le fichier nouvellement chargé.
	Upload Incoming content as File	Sélectionnez cette option si vous chargez des données provenant d'un flux d'entrée du composant précédent.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton **[...]** à côté de **Edit schema** afin de voir le schéma prédéfini ayant un champ nommé **content** décrivant les données à charger. La liste **Schema** et le bouton **[...]** à côté de **Edit schema** s'affichent uniquement lorsque l'option **Upload Incoming content as File** est sélectionnée.

	Upload Local File	Sélectionnez cette option lorsque vous chargez des données d'un fichier local.
	File	Parcourez votre système ou saisissez le chemin d'accès du fichier à charger. Ce champ s'affiche uniquement lorsque l'option **Upload Local File** est sélectionnée.
	Expose As OutputStream	Sélectionnez cette option pour exposer le flux de sortie de ce composant, pouvant être utilisé par d'autres composants pour écrire le contenu du fichier. Par exemple, vous pouvez utiliser la fonctionnalité **Use Output Stream** du composant **tFileOutputDelimited** pour alimenter un flux de sortie exposé du **tGoogleDrivePut**. Pour plus d'informations, consultez *tFileOutputDelimited*.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les informations de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone, pour charger un fichier local dans Google Drive ou encore comme composant de sortie pour charger des données dans Google Drive provenant du flux d'entrée d'un composant précédent.	
Limitation	n/a	

Gestion des fichiers avec Google Drive

Ce scénario décrit un Job chargeant deux fichiers dans un dossier vide *Talend* dans le répertoire racine de Google Drive, créant un nouveau dossier *Backup* dans le dossier *Talend*, copiant un fichier dans le dossier *Backup* et affichant tous les fichiers et dossiers dans le répertoire racine de Google Drive dans la console.

Prérequis :

1. Un compte Google a été créé et le jeton d'accès permettant l'authentification pour votre requête a été généré. Pour plus d'informations concernant l'obtention de jetons d'accès, consultez *Obtenir un jeton d'accès à Google Drive*.

2. Un dossier vide *Talend* a été créé dans le répertoire racine de Google Drive.

Ajouter et relier les composants

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette** :
   - un composant **tGoogleDriveConnection**,
   - deux **tGoogleDrivePut**,
   - un **tGoogleDriveCreate**,
   - un **tGoogleDriveCopy**,
   - un **tGoogleDriveList**,
   - un **tIterateToFlow** et
   - trois **tLogRow**.
2. Reliez le **tGoogleDriveCreate** au premier **tLogRow** à l'aide d'un lien **Row** > **Main**.
3. Reliez le **tGoogleDriveCopy** au deuxième **tLogRow** à l'aide d'un lien **Row** > **Main**.
4. Reliez le **tGoogleDriveList** au **tIterateToFlow** à l'aide d'une connexion **Row** > **Iterate**.
5. Reliez le **tIterateToFlow** au troisième **tLogRow** à l'aide d'un lien **Row** > **Main**.
6. Reliez le **tGoogleDriveConnection** au premier **tGoogleDrivePut** à l'aide d'un lien **Trigger** > **On Subjob Ok**.
7. Répétez l'opération afin de relier le premier **tGoogleDrivePut** au second **tGoogleDrivePut**, reliez le second **tGoogleDrivePut** au **tGoogleDriveCreate**, reliez le **tGoogleDriveCreate** au **tGoogleDriveCopy** et le **tGoogleDriveCopy** au **tGoogleDriveList**.

Configurer les composants

Ouvrir une connexion à Google Drive

1. Double-cliquez sur le **tGoogleDriveConnection** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **Application Name**, saisissez le nom de l'application requis par Google Drive pour accéder à son API. Dans cet exemple, saisissez *TalendProduct*.

3. Dans le champ **Access Token**, saisissez le jeton d'accès requis par Google Drive pour accéder au compte Google Drive.

Charger des fichiers dans Google Drive

1. Double-cliquez sur le premier **tGoogleDrivePut** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Use Existing Connection** et, dans la liste qui s'affiche, sélectionnez le composant configuré afin d'ouvrir votre connexion Google Drive. Dans cet exemple, saisissez *tGoogleDriveConnection_1*.

3. Sélectionnez l'option **Upload Local File** et, dans le champ **File**, parcourez votre système ou saisissez le chemin d'accès au fichier à charger. Dans cet exemple, le chemin est le suivant : *C:/Talend/Talend Release Plan.csv*.

4. Dans le champ **File Title**, saisissez le nom du fichier après chargement. Dans cet exemple, saisissez *Talend Release Plan.csv*.

5. Dans le champ **Destination Folder Name**, saisissez le nom du dossier dans lequel le fichier sera stocké dans Google Drive. Dans cet exemple, saisissez *Talend*.

6. Double-cliquez sur le second **tGoogleDrivePut** pour ouvrir sa vue **Basic settings**.

7. Répétez les étapes de Étape 2 à Étape 5 pour charger un autre fichier local dans le dossier *Talend* de Google Drive. Dans cet exemple, saisissez *Talend Products.txt*.

Créer un nouveau dossier dans Google Drive

1. Double-cliquez sur le **tGoogleDriveCreate** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Use Existing Connection** et, dans la liste qui s'affiche, sélectionnez le composant configuré pour ouvrir votre connexion à Google Drive.

3. Dans le champ **Parent Folder Name**, saisissez le nom du dossier dans lequel créer un nouveau dossier. Dans cet exemple, saisissez *Talend*.

4. Dans le champ **New Folder Name**, saisissez le nom du dossier à créer. Dans cet exemple, saisissez *Backup*.

5. Cliquez sur le bouton **[...]** à côté de **Edit schema** pour voir le schéma prédéfini du **tGoogleDriveCreate**.

6. Double-cliquez sur le premier **tLogRow** pour ouvrir sa vue **Basic settings**.

7. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Copier un fichier dans le nouveau dossier

1. Double-cliquez sur le **tGoogleDriveCopy** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Use Existing Connection** et, dans la liste qui s'affiche, sélectionnez le composant configuré pour ouvrir votre connexion à Google Drive.

3. Dans le champ **File Name**, saisissez le nom du fichier à copier. Dans cet exemple, saisissez *Talend Release Plan.csv*.

4. Dans le champ **Destination Folder Name**, saisissez le nom du dossier dans lequel copier le fichier. Dans cet exemple, saisissez *Backup*.

5. Cochez la case **Rename (set new title)** et, dans le champ **Destination Name** qui s'affiche, saisissez un nouveau nom pour le fichier après copie dans le dossier de destination. Dans cet exemple, saisissez *Talend Release Plan v1.0.csv*.

6. Cliquez sur le bouton **[...]** à côté de **Edit schema** pour voir le schéma prédéfini du **tGoogleDriveCopy**.

7. Double-cliquez sur le second **tLogRow** pour ouvrir sa vue **Basic settings**.

8. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Lister les fichiers et dossiers dans Google Drive

1. Double-cliquez sur le **tGoogleDriveList** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Use Existing Connection** et, dans la liste qui s'affiche, sélectionnez le composant configuré pour ouvrir votre connexion à Google Drive.

3. Dans le champ **Folder Name**, saisissez le nom du dossier à itérer. Dans cet exemple, utilisez le dossier racine de Google Drive. Vous pouvez utiliser l'alias *root* pour y faire référence.

4. Sélectionnez **Both** dans la liste **FileList Type** afin de lister les fichiers et dossier dans le répertoire racine.

   Cochez la case **Includes subdirectories** afin d'itérer également sur les sous-répertoires du répertoire racine.

5. Double-cliquez sur le **tIterateToFlow** pour ouvrir sa vue **Basic settings**.

6. Cliquez sur le bouton **[...]** à côté de **Edit schema** pour ouvrir l'éditeur du schéma.

   Cliquez quatre fois sur le bouton **[+]** pour ajouter quatre colonnes : *ID*, *Title* et *MimeType* de type **String** et *Size* de type **Long**.

Cliquez sur **OK** afin de valider les modifications et acceptez la propagation proposée par la boîte de dialogue.

7. Dans la table **Mapping**, renseignez la valeur pour chaque colonne en sélectionnant la cellule correspondante dans la colonne **Value** et en appuyant sur les touches **Ctrl+Espace**. Dans la liste des variables qui s'affiche, sélectionnez la variable globale correspondant au composant **tGoogleDriveList**. Dans cet exemple, sélectionnez :
   - *((String)globalMap.get("tGoogleDriveList_1_ID"))* pour la colonne *ID*,
   - *((String)globalMap.get("tGoogleDriveList_1_TITLE"))* pour la colonne *Title*,
   - *((String)globalMap.get("tGoogleDriveList_1_MIME_TYPE"))* pour la colonne *MimeType* et
   - *((Long)globalMap.get("tGoogleDriveList_1_SIZE"))* pour la colonne *Size*.
8. Double-cliquez sur le troisième **tLogRow** pour ouvrir sa vue **Basic settings**.

9. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Exécutez le Job en appuyant sur **F6** ou en cliquant sur l'onglet **Run** dans l'onglet **Run**.

```
[statistics] connecting to socket on port 3625
[statistics] connected
.----------------------------+----------------------------.
|                        tLogRow_1                         |
|=----------------------------+----------------------------=|
|parentFolderID              |newFolderID                 |
|=----------------------------+----------------------------=|
|0B2ZtkTBM5ALpaXFzNXpMOG4xM00|0B2ZtkTBM5ALpejFFUFJTQ1RQdTQ|
'----------------------------+----------------------------'
.----------------------------+----------------------------.
|                        tLogRow_2                         |
|=----------------------------+----------------------------=|
|sourceFileID                |destinationFileID           |
|=----------------------------+----------------------------=|
|0B2ZtkTBM5ALpYm4xT1FTcTMxOWc|0B2ZtkTBM5ALpbnJQRHVDQ1RRYW8|
'----------------------------+----------------------------'
.----------------------------+-----------------------------+----------------------------------+-----.
|                                          tLogRow_3                                                |
|=----------------------------+-----------------------------+----------------------------------+----=|
|ID                          |Title                        |MimeType                          |Size |
|=----------------------------+-----------------------------+----------------------------------+----=|
|0B2ZtkTBM5ALpaXFzNXpMOG4xM00|Talend                       |application/vnd.google-apps.folder|null |
|0B2ZtkTBM5ALpejFFUFJTQ1RQdTQ|Backup                       |application/vnd.google-apps.folder|null |
|0B2ZtkTBM5ALpMnJRZkc1RE9NS2c|Talend Products.txt          |text/plain                        |107  |
|0B2ZtkTBM5ALpYm4xT1FTcTMxOWc|Talend Release Plan.csv      |text/csv                          |15325|
|0B2ZtkTBM5ALpbnJQRHVDQ1RRYW8|Talend Release Plan v1.0.csv |text/csv                          |15325|
'----------------------------+-----------------------------+----------------------------------+-----'

[statistics] disconnected
```

Comme affiché dans la capture d'écran ci-dessus, deux fichiers *Talend Products.txt* et *Talend Release Plan.csv* ont été chargés dans le dossier *Talend*. Un nouveau dossier *Backup* a été créé dans le dossier *Talend* et le fichier *Talend Release Plan.csv* a été copié dans le nouveau dossier et renommé *Talend Release Plan v1.0.csv*. Finalement, les fichiers et dossiers sont listés et affichés dans la console.

tGSBucketCreate

Le composant **tGSBucketCreate** appartient à deux familles : Big Data et Cloud. Pour plus d'informations, consultez *tGSBucketCreate*.

tGSBucketDelete

Le composant **tGSBucketDelete** appartient à deux familles : Big Data et Cloud. Pour plus d'informations, consultez *tGSBucketDelete*.

tGSBucketExist

Le composant **tGSBucketExist** appartient à deux familles : Big Data et Cloud. Pour plus d'informations, consultez *tGSBucketExist*.

tGSBucketList

Le composant **tGSBucketList** appartient à deux familles : Big Data et Cloud. Pour plus d'informations, consultez *tGSBucketList*.

tGSClose

Le composant **tGSClose** appartient à deux familles : Big Data et Cloud. Pour plus d'informations, consultez *tGSClose*.

tGSConnection

Le composant **tGSConnection** appartient à deux familles : Big Data et Cloud. Pour plus d'informations, consultez *tGSConnection*.

tGSCopy

Le composant **tGSCopy** appartient à deux familles : Big Data et Cloud. Pour plus d'informations, consultez *tGSCopy*.

tGSDelete

Le composant **tGSDelete** appartient à deux familles : Big Data et Cloud. Pour plus d'informations, consultez *tGSDelete*.

tGSGet

Le composant **tGSGet** appartient à deux familles différentes : Big Data et Cloud. Pour plus d'informations, consultez *tGSGet*.

tGSList

Le composant **tGSList** appartient à deux familles différentes : Big Data et Cloud. Pour plus d'informations, consultez *tGSList*.

tGSPut

Le composant **tGSPut** appartient à deux familles différentes : Big Data et Cloud. Pour plus d'informations, consultez *tGSPut*.

tMarketoInput

Le composant **tMarketoInput** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tMarketoInput*.

tMarketoListOperation

Le composant **tMarketoListOperation** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tMarketoListOperation*.

tMarketoOutput

Le composant **tMarketoOutput** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tMarketoOutput*.

tNetsuiteInput

Le composant **tNetsuiteInput** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations concernant ce composant, *tNetsuiteInput*.

tNetsuiteOutput

Le composant **tNetsuiteOutput** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations concernant ce composant, consultez *tNetsuiteOutput*.

tS3BucketCreate

Propriétés du tS3BucketCreate

Famille de composant	Cloud/AmazonS3	
Fonction	Le composant **tS3BucketCreate** crée un bucket dans Amazon S3.	
Objectif	Le **tS3BucketCreate** est conçu pour créer un bucket dans Amazon S3.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key	L'ID de la clé d'accès (Access Key) identifie uniquement un compte AWS. Pour plus d'informations concernant la clé d'accès et sa sécurité, consultez le site Getting Your AWS Access Keys (en anglais).
	Access Secret	La Secret Access Key, combinée à la clé d'accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Region	Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (*"us-east-1"* par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Regions and Endpoints (en anglais).
	Bucket	Nom du bucket à créer.
	Die on error	Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Config client*	Cochez cette case pour configurer des paramètres client. **Client parameter** : sélectionnez les paramètres client dans la liste. **Value** : saisissez la valeur des paramètres. Cette option n'est pas disponible lorsque la case **Use existing connection** est cochée.
	tStatCatcher Statistics	Cochez cette case afin de collecter les données de log au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone ou avec d'autres composants S3, par exemple le **tS3BucketExist**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé au **tS3BucketCreate**, consultez *Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3* .

tS3BucketDelete

tS3BucketDelete properties

Famille de composant	Cloud/AmazonS3	
Fonction	Le composant **tS3BucketDelete** supprime un bucket vide d'Amazon S3.	
Objectif	Le composant **tS3BucketDelete** est conçu pour supprimer un bucket vide d'Amazon S3.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key	L'ID de la clé d'accès (Access Key) identifie uniquement un compte AWS. Pour plus d'informations concernant la clé d'accès et sa sécurité, consultez le site Getting Your AWS Access Keys (en anglais).
	Access Secret	La Secret Access Key, combinée à la clé d'accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Region	Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (*"us-east-1"* par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Regions and Endpoints (en anglais).
	Bucket	Nom du bucket à supprimer.
	Die on error	Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Config client*	Cochez cette case pour configurer des paramètres client. **Client parameter** : sélectionnez les paramètres client dans la liste. **Value** : saisissez la valeur des paramètres. Cette option n'est pas disponible lorsque la case **Use existing connection** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone ou avec d'autres composants S3, par exemple le **tS3BucketList**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé au **tS3BucketDelete**, consultez *Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3* .

tS3BucketExist

Propriétés du tS3BucketExist

Famille de composant	Cloud/AmazonS3	
Fonction	Le composant **tS3BucketExist** vérifie qu'un bucket existe dans Amazon S3.	
Objectif	Le composant **tS3BucketExist** est conçu pour vérifier que le bucket spécifié existe dans Amazon S3.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key	L'ID de la clé d'accès (Access Key) identifie uniquement un compte AWS. Pour plus d'informations concernant la clé d'accès et sa sécurité, consultez le site Getting Your AWS Access Keys (en anglais).
	Access Secret	La Secret Access Key, combinée à la clé d'accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Region	Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (*"us-east-1"* par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Regions and Endpoints (en anglais).
	Bucket	Nom du bucket sur le serveur S3.
	Die on error	Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Config client*	Cochez cette case pour configurer des paramètres client. **Client parameter** : sélectionnez les paramètres client dans la liste. **Value** : saisissez la valeur des paramètres. Cette option n'est pas disponible lorsque la case **Use existing connection** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**BUCKET_EXIST** : présence ou non d'un bucket spécifié. Cette variable est une variable **Flow** et retourne un booléen.	

	BUCKET_NAME : nom d'un bucket spécifié. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé avec d'autres composants S3, comme le **tS3BucketCreate**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3

Dans ce scénario, le **tS3BucketExist** est utilisé pour vérifier l'absence d'un bucket. Le composant **tS3BucketCreate** est utilisé pour créer ce bucket après confirmation de l'absence du bucket. Le **tS3BucketList** permet de lister tous les buckets dans Amazon S3.

Construire le Job

1. Déposez un **tS3Connection**, un **tS3BucketExist**, un **tS3BucketCreate**, un **tS3BucketList**, un **tIterateToFlow** et un **tLogRow** dans l'espace de modélisation graphique.

2. Reliez le **tS3Connection** au **tS3BucketExist** à l'aide d'un lien **OnSubjobOk**.

3. Reliez le **tS3BucketExist** au **tS3BucketCreate** à l'aide d'un lien **Run if**.

4. Reliez le **tS3BucketCreate** au **tS3BucketList** à l'aide d'un lien **OnSubjobOk**.

5. Reliez le **tS3BucketList** au **tIterateToFlow** à l'aide d'un lien **Row > Iterate**.

6. Reliez le **tIterateToFlow** au **tLogRow** à l'aide d'un lien **Row > Main**.

Configurer les composants

1. Double-cliquez sur le **tS3Connection** pour ouvrir sa vue **Basic settings**.

2. Dans les champ **Access Key** et **Secret Key**, saisissez vos informations d'authentification.

3. Double-cliquez sur le composant **tS3BucketExist** pour ouvrir sa vue **Basic settings**.

4. Cochez la case **Use existing connection** afin de réutiliser la connexion définie dans le composant précédent.

5. Dans le champ **Bucket**, saisissez le nom du bucket afin de vérifier son existence.

6. Double-cliquez sur le lien **If** afin de configurer sa condition.

7. Dans le champ **Condition**, saisissez l'expression :

```
!((Boolean)globalMap.get("tS3BucketExist_1_BUCKET_EXIST"))
```

Ainsi, le reste du Job est exécuté si le bucket spécifié n'existe pas.

8. Double-cliquez sur le **tS3BucketCreate** pour ouvrir sa vue **Basic settings**.

Cochez la case **Use existing connection** afin de réutiliser la connexion.

Dans le champ **Bucket**, saisissez le nom du bucket à créer.

9. Double-cliquez sur le **tS3BucketList** pour ouvrir sa vue **Basic settings**.

Cochez la case **Use existing connection** pour réutiliser la connexion.

10. Double-cliquez sur le **tIterateToFlow** pour ouvrir sa vue **Basic settings**.

11. Cliquez sur le bouton **Edit schema** pour ouvrir l'éditeur de schéma.

Cliquez sur le bouton **[+]** pour ajouter une colonne. Nommez-la *bucket_list* et définissez son type à **String**.

Cliquez sur **OK** pour valider la configuration et fermer l'éditeur.

12. Dans la zone **Mapping**, appuyez sur les touches **Ctrl+Espace** dans le champ **Value** afin de sélectionner la variable *tS3BucketList_1_CURRENT_BUCKET_NAME*.

13. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

Sélectionnez l'option **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter le Job.

```
[statistics] connecting to socket on port 3661
[statistics] connected
.-----------.
| tLogRow_1 |
|=----------=|
|bucket_list|
|=----------=|
|movies_andy|
|songs-andy |
'-----------'

[statistics] disconnected
```

Comme affiché ci-dessus, le bucket est créé et tous les buckets sont listés.

3. Allez dans la console Web S3 :

Vous pouvez constater que le bucket a bien été créé sur le serveur S3.

4. Rafraîchissez le client S3 Browser :

La capture d'écran ci-dessus vous montre que l'action S3 Create s'est déroulée avec succès.

tS3BucketList

Propriétés du tS3BucketList

Famille de composant	Cloud/AmazonS3	
Fonction	Le composant **tS3BucketList** liste tous les buckets dans Amazon S3.	
Objectif	Le **tS3BucketList** est conçu pour lister tous les buckets dans Amazon S3.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key	L'ID de la clé d'accès (Access Key) identifie uniquement un compte AWS. Pour plus d'informations concernant la clé d'accès et sa sécurité, consultez le site Getting Your AWS Access Keys (en anglais).
	Access Secret	La Secret Access Key, combinée à la clé d'accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Region	Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (*"us-east-1"* par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Regions and Endpoints (en anglais).
	Die on error	Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Config client*	Cochez cette case pour configurer des paramètres client. **Client parameter** : sélectionnez les paramètres client dans la liste. **Value** : saisissez la valeur des paramètres. Cette option n'est pas disponible lorsque la case **Use existing connection** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**CURRENT_BUCKET_NAME** : nom du bucket courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_BUCKET** : nombre de buckets. Cette variable est une variable **After** et retourne un nombre entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone ou avec d'autres composants S3, comme le **tS3BucketDelete**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé au **tS3BucketList**, consultez *Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3* .

tS3Close

Propriétés du tS3Close

Famille de composant	Cloud/AmazonS3	
Fonction	Le composant tS3Close permet de fermer une connexion à Amazon S3.	
Objectif	Le **tS3Close** est conçu pour fermer une connexion à Amazon S3, libérant ainsi les ressources réseau.	
Basic settings	*Component List*	Sélectionnez dans la liste le composant de connexion.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	En tant que composant de sortie, ce composant doit être utilisé avec d'autres composants S3, par exemple le **tS3Connection**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au **tS3Close**, consultez *Scénario : Lister les fichiers d'un bucket ayant le même préfixe*.

tS3Connection

Propriétés du tS3Connection

Famille de composant	Cloud/AmazonS3	
Fonction	Le composant tS3Connection permet de vous connecter à Amazon S3.	
Objectif	Le composant **tS3Connection** est conçu pour vous connecter à Amazon S3 afin de stocker et de récupérer des données.	
Basic settings	*Access Key*	L'ID de la clé d'accès (Access Key) identifie uniquement un compte AWS. Pour plus d'informations concernant la clé d'accès et sa sécurité, consultez le site Getting Your AWS Access Keys (en anglais).
	Access Secret	La Secret Access Key, combinée à la clé d'accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Region	Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (*"us-east-1"* par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Regions and Endpoints (en anglais).
Advanced settings	*Config client*	Cochez cette case pour configurer des paramètres client. **Client parameter** : sélectionnez les paramètres client dans la liste. **Value** : saisissez la valeur des paramètres.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est un composant de début, il doit être utilisé avec d'autres composants S3.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé contenant un composant **tS3Connection**, consultez *Scénario : Echange de fichiers avec Amazon S3* .

tS3Delete

Propriétés du tS3Delete

Famille de composant	Cloud/AmazonS3	
Fonction	Le composant **tS3Delete** permet de supprimer un fichier dans Amazon S3.	
Objectif	Le **tS3Delete** est conçu pour supprimer un fichier dans Amazon S3.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key	L'ID de la clé d'accès (Access Key) identifie uniquement un compte AWS. Pour plus d'informations concernant la clé d'accès et sa sécurité, consultez le site Getting Your AWS Access Keys (en anglais).
	Access Secret	La Secret Access Key, combinée à la clé d'accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Region	Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (*"us-east-1"* par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Regions and Endpoints (en anglais).
	Bucket	Nom du bucket dans Amazon S3.
	Key	Nom du fichier à supprimer.
	Die on error	Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Config client*	Cochez cette case pour configurer des paramètres client. **Client parameter** : sélectionnez les paramètres client dans la liste. **Value** : saisissez la valeur des paramètres. Cette option n'est pas disponible lorsque la case **Use existing connection** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone ou avec d'autres composants S3, par exemple le **tS3BucketList**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé au composant **tS3Delete**, consultez *Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3* .

tS3Get

Propriétés du tS3Get

Famille de composant	Cloud/AmazonS3	
Fonction	Ce composant récupère un fichier d'Amazon S3.	
Objectif	Le composant **tS3Get** est conçu pour récupérer un fichier d'Amazon S3.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key	L'ID de la clé d'accès (Access Key) identifie uniquement un compte AWS. Pour plus d'informations concernant la clé d'accès et sa sécurité, consultez le site Getting Your AWS Access Keys (en anglais).
	Access Secret	La Secret Access Key, combinée à la clé d'accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Region	Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (*"us-east-1"* par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Regions and Endpoints (en anglais).
	Bucket	Nom du bucket du serveur S3.
	Key	Nom du fichier à sauvegarder sur le serveur S3.
	File	Nom du fichier local à charger sur le serveur S3.
	Die on error	Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Config client*	Cochez cette case pour configurer des paramètres client. **Client parameter** : sélectionnez les paramètres client dans la liste. **Value** : saisissez la valeur des paramètres. Cette option n'est pas disponible lorsque la case **Use existing connection** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone ou avec d'autres composants S3, comme le **tS3Connection**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé utilisant le composant **tS3Get**, consultez *Scénario : Echange de fichiers avec Amazon S3* .

tS3List

Propriétés du tS3List

Famille de composant	Cloud/AmazonS3	
Fonction	Le composant **tS3List** permet de lister les fichiers dans Amazon S3 selon les paramètres de préfixe du bucket ou du fichier.	
Objectif	Le **tS3List** est conçu pour lister les fichiers dans Amazon S3 selon les paramètres de filtre sur le bucket ou le fichier définis.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key	L'ID de la clé d'accès (Access Key) identifie uniquement un compte AWS. Pour plus d'informations concernant la clé d'accès et sa sécurité, consultez le site Getting Your AWS Access Keys (en anglais).
	Access Secret	La Secret Access Key, combinée à la clé d'accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Region	Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (*"us-east-1"* par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Regions and Endpoints (en anglais).
	List all bucket objects	Cochez cette case pour lister tous les fichiers sur le serveur S3. **Key prefix** : saisissez le préfixe des fichiers à lister. Ainsi, seuls les fichiers contenant ce préfixe seront listés.
	Bucket	Cliquez sur le bouton **[+]** pour ajouter une ou plusieurs lignes afin de définir les préfixes des buckets et fichiers. **Bucket name** : nom du bucket duquel lister les fichiers. **Key prefix** : préfixe des fichiers à lister. Cette option est indisponible lorsque la case **List all bucket objects** est cochée.
	Die on error	Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Config client*	Cochez cette case pour configurer des paramètres client. **Client parameter** : sélectionnez les paramètres client dans la liste. **Value** : saisissez la valeur des paramètres. Cette option n'est pas disponible lorsque la case **Use existing connection** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données	

	parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**CURRENT_BUCKET** : nom du bucket courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_KEY** : clé courante. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_BUCKET** : nombre de buckets. Cette variable est une variable **After** et retourne un nombre entier. **NB_BUCKET_OBJECT** : nombre d'objets dans tous les buckets. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone ou avec d'autres composants S3, comme le **tS3Delete**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Lister les fichiers d'un bucket ayant le même préfixe

Dans ce scénario, le **tS3List** est utilisé pour lister tous les fichiers d'un bucket ayant le même préfixe.

Voici les fichiers du bucket :

Pour plus d'informations concernant la création d'un bucket et comment y mettre des fichiers, consultez *Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3* et *Scénario : Echange de fichiers avec Amazon S3* .

Construire le Job

1. De la **Palette**, déposez un composant **tS3Connection**, un **tS3List**, un **tIterateToFlow**, un **tLogRow** et un **tS3Close** dans l'espace de modélisation graphique.
2. Reliez le **tS3Connection** au **tS3List** à l'aide d'un lien **OnSubjobOk**.
3. Reliez le **tS3List** au **tIterateToFlow** à l'aide d'un lien **Row > Iterate**.
4. Reliez le **tIterateToFlow** au **tLogRow** à l'aide d'un lien **Row > Main**.
5. Reliez le **tS3List** au **tS3Close** à l'aide d'un lien **OnSubjobOk**.

Configurer les composants

1. Double-cliquez sur le **tS3Connection** pour ouvrir sa vue **Basic settings**.

2. Dans les champs **Access Key** et **Secret Key**, saisissez les informations d'authentification.

3. Double-cliquez sur le **tS3List** pour ouvrir sa vue **Basic settings**.

4. Cochez la case **Use existing connection** afin de réutiliser la connexion définie précédemment.

5. Dans la table **Bucket**, cliquez sur le bouton **[+]** pour ajouter une ligne.

6. Dans les champs **Bucket name** et **Key prefix**, saisissez le nom du bucket et le préfixe du fichier.

   Ainsi, seuls les fichiers ayant le préfixe spécifié seront listés.

7. Double-cliquez sur le **tIterateToFlow** pour ouvrir sa vue **Basic settings**.

8. Cliquez sur **Edit schema** pour ouvrir l'éditeur du schéma.

Cliquez sur le bouton **[+]** pour ajouter une colonne. Nommez-la *file_list* et définissez son type à **String**.

Cliquez sur **OK** pour valider la configuration et fermer l'éditeur.

9. Dans la table **Mapping**, appuyez sur les touches **Ctrl+Espace** dans la colonne **Value** afin de sélectionner la variable *tS3List_1_CURRENT_KEY*.

10. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

Sélectionnez l'option **Table (print values in cells of a table)** pour un affichage optimal des résultats.

11. Double-cliquez sur le composant **tS3Close** afin d'ouvrir sa vue **Basic settings**.

Comme un seul composant de connexion est utilisé dans ce Job, vous n'avez pas besoin de le sélectionner dans la liste **Component List**, il est sélectionné par défaut.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur la touche **F6** pour exécuter le Job.

```
[statistics] connecting to socket on port 3704
[statistics] connected
.---------.
|tLogRow_1|
|=--------=|
|file_list|
|=--------=|
|in-01.xls|
|in-02.xls|
'---------'

[statistics] disconnected
```

Comme affiché ci-dessus, seuls les fichiers ayant le préfixe *"in"* sont listés.

tS3Put

Propriétés du tS3Put

Famille de composant	Cloud/AmazonS3	
Fonction	Met un fichier dans Amazon S3.	
Objectif	Le composant **tS3Put** est conçu pour mettre un fichier dans Amazon S3 afin d'y stocker les données.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Access Key	L'ID de la clé d'accès (Access Key) identifie uniquement un compte AWS. Pour plus d'informations concernant la clé d'accès et sa sécurité, consultez le site Getting Your AWS Access Keys (en anglais).
	Access Secret	La Secret Access Key, combinée à la clé d'accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Region	Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (*"us-east-1"* par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Regions and Endpoints (en anglais).
	Bucket	Nom du bucket du serveur S3.
	Key	Nom du fichier à sauvegarder sur le serveur S3.
	File	Nom du fichier local à charger sur le serveur S3.
	Die on error	Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Config client*	Cochez cette case pour configurer des paramètres client. **Client parameter** : sélectionnez les paramètres client dans la liste. **Value** : saisissez la valeur des paramètres. Cette option n'est pas disponible lorsque la case **Use existing connection** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant peut être utilisé en standalone ou avec d'autres composants S3, par exemple le **tS3Connection**.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Echange de fichiers avec Amazon S3

Dans ce scénario, le composant **tS3Put** est utilisé pour envoyer un fichier sur le serveur Amazon S3 pendant que le **tS3Get** récupère ce fichier.

Construire le Job

1. Déposez un **tS3Connection**, un **tS3Put** et un **tS3Get** de la **Palette** dans l'espace de modélisation graphique.
2. Reliez le **tS3Connection** au **tS3Put** à l'aide d'un lien **OnSubjobOk**.
3. Reliez le **tS3Put** au **tS3Get** à l'aide d'un lien **OnSubjobOk**.

Configurer les composants

1. Double-cliquez sur le **tS3Connection** pour ouvrir sa vue **Basic settings**.

2. Dans les champs **Access Key** et **Secret Key**, saisissez les informations d'authentification.

3. Double-cliquez sur le **tS3Put** pour ouvrir sa vue **Basic settings**.

4. Cochez la case **Use existing connection** pour réutiliser la connexion.

5. Dans le champ **Bucket**, saisissez le nom du bucket.

6. Dans le champ **Key**, saisissez le nom du fichier à sauvegarder sur le serveur S3.

7. Dans le champ **File**, saisissez le nom du fichier à charger dans le serveur S3.

8. Double-cliquez sur le composant **tS3Get** pour ouvrir sa vue **Basic settings**.

9. Cochez la case **Use existing connection** pour réutiliser la connexion.

10. Dans le champ **Bucket**, saisissez le nom du bucket.

11. Dans le champ **Key**, saisissez le nom du fichier à récupérer du serveur S3.

12. Dans le champ **File**, saisissez le nom local du fichier récupéré.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter le Job.

```
Starting job s3_put_get at 14:22 08/10/2013.

[statistics] connecting to socket on port 3950
[statistics] connected
[statistics] disconnected

Job s3_put_get ended at 14:22 08/10/2013. [exit code=0]
```

3. Consultez la console Web S3 :

Comme affiché ci-dessus, le fichier a bien été mis sur le serveur S3.

4. Rafraîchissez le client S3 Browser :

Comme affiché ci-dessus, cela montre également l'action réussie du composant S3 Put.

5. Allez dans le dossier où est sauvegardé localement le fichier récupéré :

Comme affiché ci-dessus, le fichier distant est récupéré dans le disque local, ce qui prouve que l'action Get S3 s'est bien déroulée.

tSalesforceBulkExec

Le composant **tSalesforceBulkExec** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSalesforceBulkExec*.

tSalesforceConnection

Le composant **tSalesforceConnection** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSalesforceConnection*.

tSalesforceGetDeleted

Le composant **tSalesforceGetDeleted** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSalesforceGetDeleted*.

tSalesforceGetServerTimestamp

Le composant **tSalesforceGetServerTimestamp** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSalesforceGetServerTimestamp*.

tSalesforceGetUpdated

Le composant **tSalesforceGetUpdated** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSalesforceGetUpdated*.

tSalesforceInput

Le composant **tSalesforceInput** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSalesforceInput*.

tSalesforceOutput

Le composant **tSalesforceOutput** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSalesforceOutput*.

tSalesforceOutputBulk

Le composant **tSalesforceOutputBulk** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSalesforceOutputBulk*.

tSalesforceOutputBulkExec

Le composant **tSalesforceOutputBulkExec** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSalesforceOutputBulkExec*.

tSalesforceWaveBulkExec

Le **SalesforceWaveBulkExec** appartient à deux familles : Business et Cloud. Pour plus d'informations, consultez *tSalesforceWaveBulkExec*.

tSalesforceWaveOutputBulkExec

Le **tSalesforceWaveOutputBulkExec** appartient à deux familles : Business et Cloud. Pour plus d'informations, consultez *tSalesforceWaveOutputBulkExec*.

tSugarCRMInput

Le composant **tSugarCRMInput** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSugarCRMInput*.

tSugarCRMOutput

Le composant **tSugarCRMOutput** appartient à deux familles différentes : Business et Cloud. Pour plus d'informations, consultez *tSugarCRMOutput*.

Composants Custom Code

Ce chapitre passe en revue les principaux composants présents dans la famille **Custom Code** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **Custom Code** regroupe tous les composants répondant à un besoin spécifique de création de code à la volée.

tGroovy

Propriétés du tGroovy

Famille de composant	Custom Code	
Fonction	Le **tGroovy** permet de saisir du code personnalisé afin de l'intégrer dans le programme **Talend**. Ce code est exécuté une seule fois.	
Objectif	Le **tGroovy** permet d'étendre les fonctionnalités du Job **Talend** grâce au langage Groovy, qui est une syntaxe simplifiée du Java.	
Basic settings	*Groovy Script*	Saisissez le code Groovy que vous souhaitez exécuter.
	Variables	Ce tableau comprend deux colonnes : **Name** : nom de la variable appelée dans le code. **Value** : valeur associée à cette variable.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone, ou en tant que sous-job à un seul composant.	
Limitation	Il est nécessaire de connaître le langage Groovy.	

Scénarios associés

- Pour un scénario utilisant du Code Groovy, consultez *Scénario : Appeler du code Groovy contenu dans un fichier*.
- Pour un exemple d'utilisation proche dans le mode de fonctionnement, consultez *Scénario : Imprimer le contenu d'une variable* du composant **tJava**.

tGroovyFile

Propriétés du tGroovyFile

Famille de composant	Custom Code	
Fonction	Le **tGroovyFile** permet d'appeler un script existant développé en Groovy.	
Objectif	Le **tGroovyFile** permet d'étendre les fonctionnalités du Job **Talend** grâce au langage Groovy, qui est une syntaxe simplifiée du Java.	
Basic settings	*Groovy File*	Nom et chemin d'accès au fichier contenant votre code Groovy.
	Variables	Ce tableau comprend deux colonnes : **Name** : nom de la variable appelée dans le code. **Value** : valeur associée à cette variable.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone, ou en tant que sous-job à un seul composant.	
Limitation	Il est nécessaire de connaître le langage Groovy.	

Scénario : Appeler du code Groovy contenu dans un fichier

Ce scénario utilise un composant **tGroovyFile** en standalone. Le Job appelle un fichier contenant du code Groovy afin d'afficher les informations de ce fichier dans la **Console**.

Construire le Job

Développez le dossier **Custom_Code** de la **Palette** et déposez un composant **tGroovyFile** dans l'espace de modélisation graphique.

Configurer le composant tGroovyFile

1. Double-cliquez sur le composant pour ouvrir sa vue **Component**.

2. Dans le champ **Groovy File**, saisissez le chemin d'accès au fichier contenant le code Groovy, ou parcourez votre système jusqu'à ce fichier. Dans cet exemple, le chemin est : *D:/Input/Ageducapitaine.txt*. Le fichier contient le code Groovy suivant :

```
println("The captain is " + age +  " years old")
```

3. Dans la table **Variables**, ajoutez une ligne en cliquant sur le bouton **[+]**.

4. Dans la colonne **Name**, saisissez *"age"* puis, dans la colonne **Value**, saisissez *50*.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

   La **Console** affiche les informations contenues dans le fichier d'entrée, auxquelles est ajouté le résultat de la variable.

```
Starting job tGroovyFile at 09:49 17/02/2010.

[statistics] connecting to socket on port 4016
[statistics] connected
The captain is 50 years old
[statistics] disconnected
Job tGroovyFile ended at 09:49 17/02/2010. [exit code=0]
```

tJava

Propriétés du tJava

Famille de composant	Custom Code	
Fonction	Le **tJava** permet de saisir du code personnalisé afin de l'intégrer dans le programme **Talend**. Ce code est exécuté une seule fois.	
Objectif	Le **tJava** permet d'étendre les fonctionnalités du Job **Talend** grâce au langage Java.	
Basic settings	*Code*	Saisissez le code Java que vous souhaitez exécuter. Pour plus d'informations concernant la syntaxe des fonctions Java spécifiques à **Talend**, consultez l'aide en ligne du *Studio Talend* dans le menu Help Contents > Developer Guide > API Reference. Pour la référence Java complète, consultez http://docs.oracle.com/javaee/6/api/ (en anglais).
Advanced settings	*Import*	Saisissez le code Java permettant d'importer, si nécessaire, la librairie externe utilisée dans le champ **Main code** de l'onglet **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que sous-job à un seul composant.	
Limitation	Il est nécessaire de connaître le langage Java.	

Scénario : Imprimer le contenu d'une variable

Le scénario suivant est une simple démonstration de l'application possible du composant **tJava**. L'objectif de ce Job est d'imprimer les lignes traitées à l'aide d'une commande Java et des variables globales fournies dans le *Studio Talend*.

Construire le Job

1. Cliquez et déposez les composants dans l'espace de modélisation : **tFileInputDelimited**, **tFileOutputExcel**, **tJava**.

2. Connectez le composant **tFileInputDelimited** au **tFileOutputExcel** à l'aide d'une connexion de type **Row Main**. Le contenu du fichier .txt délimité sera transmis à un fichier de type XSL via cette connexion sans subir de transformation.

3. Ensuite, connectez le composant **tFileInputDelimited** au composant **tJava** à l'aide d'un lien de type **Trigger** > **On Subjob Ok**. Ce lien met en place un séquençage ordonnant au **tJava** de se lancer à la fin du processus principal.

Configurer le composant d'entrée

1. Définissez les paramètres du composant **tFileInputDelimited** dans l'onglet **Basic settings** de la vue **Component**. Le fichier d'entrée utilisé en exemple est un simple fichier texte à deux colonnes : **Name** (des noms de personnes) et leur **Emails** respectifs.

2. Pour cet exemple, le schéma n'a pas été centralisé dans le référentiel, vous devez donc configurer manuellement le schéma de ces deux colonnes.

3. Cliquez sur le bouton **Edit Schema**.

4. Une boîte de dialogue vous demandant si vous souhaitez propager les modifications apportées apparaît, cliquez sur **OK** pour accepter la propagation. Ainsi, le schéma du composant **tFileOutputExcel** est automatiquement renseigné à partir du schéma du composant précédent. Vous n'aurez donc pas à définir le schéma de nouveau.

Configurer le composant de sortie

Paramétrez le fichier de sortie afin qu'il reçoive le contenu du fichier d'entrée sans modifications. Si le fichier n'existe pas déjà, il sera créé.

Dans cet exemple, dans le champ **Sheet name**, indiquez le nom de la feuille Excel, ici *Email* et la case **Header** est cochée pour indiquer que le fichier contient un en-tête.

Configurer le composant tJava

1. Sélectionnez le composant **tJava** pour définir la commande Java à exécuter.

2. Dans la zone **Code**, saisissez la commande suivante :

```
String var = "Nb of line processed: ";
var = var + globalMap.get("tFileInputDelimited_1_NB_LINE");
System.out.println(var);
```

Dans cet exemple précis, utilisez la variable **NB_Line**. Pour accéder à la liste des variables globales, appuyez sur **Ctrl + Espace** de votre clavier et sélectionnez la variable globale correspondante.

Exécuter le Job

1. Enregistrez le Job en appuyant sur **Ctrl+S**.

2. Appuyez sur **F6** pour l'exécuter.

```
Starting job JavaDb at 13:53 20/08/2007.
Nb of line processed: 4
Job JavaDb ended at 13:53 20/08/2007. [exit code=0]
```

Le contenu est donc transmis au fichier Excel spécifié et le nombre de lignes traitées est affiché dans la console **Run**.

tJavaFlex

Propriétés du tJavaFlex

Famille de composant	Custom Code	
Fonction	Le **tJavaFlex** permet de saisir du code personnalisé afin de l'intégrer dans le programme Talend. Avec le **tJavaFlex**, vous pouvez saisir les trois différentes parties du code Java (start, main et end) constituant un composant qui permet d'exécuter une opération souhaitée et non disponible dans les actuels composants **Talend**.	
Objectif	Le **tJavaFlex** vous permet d'ajouter du code Java dans ses sections de code Start/Main/End.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Data Auto Propagate	Cochez cette case pour propager automatiquement les données au composant suivant.
	Start code	Saisissez le code Java qui va être appelé lors de la phase d'initialisation.
	Main code	Saisissez le code Java qui va être appliqué pour chaque ligne du flux.
	End code	Saisissez le code Java qui va être appelé lors de la phase de clôture.

Advanced settings	*Import*	Saisissez le code Java permettant d'importer, si nécessaire, la librairie externe utilisée dans le champ **Main code** de l'onglet **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en tant que composant de début, de milieu et de fin de Job, ainsi qu'en tant que sous-job à un seul composant.	
Limitation	Il est nécessaire de connaître le langage Java.	

Scénario 1 : Générer un flux de données

Ce scénario décrit un Job à deux composants dont le but est de générer un flux de trois lignes décrivant les différentes civilité : mademoiselle (Miss), madame (Mrs) et monsieur (Mr), et de les afficher dans la console.

Construire le Job

1. Cliquez-glissez un composant **tJavaFlex** de la famille **Custom Code** et un composant **tLogRow** de la famille **Logs & Errors** dans le Job Designer.

2. Connectez-les via un lien **Row Main**.

Configurer le composant tJavaFlex

1. Double-cliquez sur le composant **tJavaFlex** pour paramétrer ses propriétés.

2. Dans la liste **Schema**, sélectionnez le mode **Built-in** et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour décrire manuellement la structure des données à passer au composant suivant.

3. Cliquez sur le bouton **[+]** pour ajouter deux colonnes : *key* de type **Integer** et *value* de type **String**.

4. Cliquez **OK** pour valider les changements et fermer la boîte de dialogue.

5. Dans les propriétés du **tJavaFlex**, cochez la case **Data Auto Propagate** pour propager automatiquement les données au composant suivant.

   Dans cet exemple, n'effectuez pas de transformation sur les données récupérées.

6. Dans le champ **Start code**, saisissez le code qui sera exécuté lors de la phase d'initialisation.

   Ici, le code indique l'initialisation du composant **tJavaFlex** en affichant le message START et met en place la boucle et les variables utilisées par la suite dans le code Java :

```
System.out.println("## START\n#");
String [] valueArray = {"Miss", "Mrs", "Mr"};

for (int i=0;i<valueArray.length;i++) {
```

7. Dans le champ **Main code**, saisissez le code qui sera appliqué pour chaque ligne de données. Ici, l'objectif est de retourner chaque clé (*key*) avec sa valeur :

```
row1.key = i;
row1.value = valueArray[i];
```

Main code

```
// Main part of your Java code
(loop)

row1.key = i;
row1.value = valueArray[i];
```

*Dans le champ **Main code**, "row1" correspond au nom du lien qui sort du **tJavaFlex**. Si vous renommez ce lien, vous devez modifier le code de ce champ.*

8. Dans le champ **End code**, saisissez le code qui sera exécuté lors de la phase de clôture. Ici, l'accolade ferme la boucle et le code indique la fin de l'exécution du **tJavaFlex** en affichant le message END :

```
}
System.out.println("#\n## END");
```

End code

```
// End part of your Java code

}
System.out.println("#\n##
END");
```

9. Si nécessaire, assurez-vous que le schéma a bien été propagé au **tLogRow** en double-cliquant sur le composant et en cliquant sur le bouton **[...]** à côté de **Edit schema**.

Sauvegarder et exécuter le Job

1. Enregistrez le Job en appuyant sur **Ctrl+S**.

2. Exécutez le Job en appuyant sur la touche **F6** ou en cliquant sur le bouton **Run**, dans l'onglet **Run**.

```
Starting job tJavaFlex_scenario1 at 14:49 02/09/2009.

## START
#
0|Miss
1|Mrs
2|Mr
#
## END
Job tJavaFlex_scenario1 ended at 14:49 02/09/2009. [exit code=0]
```

La console affiche les trois civilités, ainsi que leur clés correspondantes.

Scénario 2 : Traiter des lignes de données avec le tJavaFlex

Ce scénario décrit un Job à deux composants générant des données de manière aléatoire, récupérant et traitant ces données ligne par ligne avec du code Java via le **tJavaFlex**.

Construire le Job

1. Cliquez-glissez un composant **tRowGenerator** de la famille **Misc** et un composant **tJavaFlex** de la famille **Custom Code** dans le Job Designer.
2. Connectez-les via un lien **Row** > **Main**.

Configurer le composant d'entrée

1. Double-cliquez sur le composant **tRowGenerator** pour ouvrir son éditeur.

2. Dans l'éditeur, cliquez sur le bouton **[+]** pour ajouter quatre colonnes : *number*, *txt*, *date* et *flag*.
3. Définissez le schéma en paramétrant ces quatre colonnes conformément à la capture d'écran précédente.
4. Dans la colonne **Functions**, sélectionnez la fonction **[...]** dans la liste de chaque colonne.
5. Dans la colonne **Parameters**, saisissez 10 paramètres différents pour chaque colonne. Ces 10 paramètres correspondent aux données qui seront retournées de manière aléatoire lors de l'exécution du **tRowGenerator**.

6. Cliquez sur **OK** pour valider vos changements et fermer l'éditeur.

Configurer le composant tJavaFlex

1. Double-cliquez sur le composant **tJavaFlex** pour paramétrer ses propriétés.

2. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

3. Dans le champ **Start code**, saisissez le code qui sera exécuté lors de la phase d'initialisation. Ici, le code indique l'initialisation du composant **tJavaFlex** en affichant le message START et définit une variable qui sera utilisée par la suite dans le code Java :

```
System.out.println("## START\n#");
int i = 0;
```

4. Dans le champ **Main code**, saisissez le code qui sera appliqué pour chaque ligne de données. Ici, l'objectif est d'indiquer le numéro de la ligne en partant de 0, puis afficher le numéro et le texte aléatoire que vous mettez en majuscules et la date aléatoire définis dans le **tRowGenerator**. Ensuite, créez une condition affichant si le statut est **true** (vrai) ou **false** (faux) et incrémentez le numéro de la ligne :

```
System.out.print(" row" + i + ":");
System.out.print("# number:" + row1.number);
System.out.print (" | txt:" + row1.txt.toUpperCase());
System.out.print(" | date:" + row1.date);
if(row1.flag) System.out.println(" | flag: true");
else  System.out.println(" | flag: false");

i++;
```

*Dans le champ **Main code**, "row1" correspond au nom du lien qui sort du **tJavaFlex**. Si vous renommez ce lien, il faut penser à modifier le code de ce champ.*

5. Dans le champ **End code**, saisissez le code qui sera exécuté lors de la phase de clôture. Ici, le code indique la fin de l'exécution du composant **tJavaFlex** en affichant le message END :

```
System.out.println("#\n## END");
```

Sauvegarder et exécuter le Job

1. Enregistrez votre Job en appuyant sur **Ctrl+S**.

2. Appuyez sur **F6** ou cliquez sur **Run** dans l'onglet **Run** pour l'exécuter.

```
Starting job tJavaFlex_scenario2 at 18:35 02/09/2009.

## START
#
 row0:# number:10 | txt:TEXT5 | date:2006-05-25 | flag: false
 row1:# number:7 | txt:TEXT7 | date:2006-06-26 | flag: true
 row2:# number:5 | txt:TEXT2 | date:2006-05-25 | flag: false
 row3:# number:6 | txt:TEXT1 | date:2005-08-28 | flag: true
 row4:# number:8 | txt:TEXT9 | date:2006-05-25 | flag: false
 row5:# number:9 | txt:TEXT1 | date:2007-01-21 | flag: false
 row6:# number:7 | txt:TEXT5 | date:2004-11-21 | flag: true
 row7:# number:9 | txt:TEXT4 | date:2005-08-29 | flag: true
 row8:# number:4 | txt:TEXT6 | date:2006-06-26 | flag: false
 row9:# number:3 | txt:TEXT10 | date:2006-05-25 | flag: false
#
## END
Job tJavaFlex_scenario2 ended at 18:35 02/09/2009. [exit
code=0]
```

La console affiche les données générées aléatoirement et modifiées à l'aide du code Java dans le **tJavaFlex**.

tJavaRow

Propriétés du tJavaRow

Famille de composant	Custom Code	
Fonction	Le **tJavaRow** permet de saisir du code personnalisé afin de l'intégrer dans le programme **Talend**. Avec le **tJavaRow**, vous pouvez saisir le code Java qui va être appliqué pour chaque ligne du flux.	
Objectif	Le **tJavaRow** permet d'étendre les fonctionnalités du Job **Talend** grâce au langage Java.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
	Code	Saisissez le code Java qui va être appliqué pour chaque ligne du flux.
Advanced settings	*Import*	Saisissez le code Java permettant d'importer, si nécessaire, la librairie externe utilisée dans le champ **Main code** de l'onglet **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Pour saisir une variable globale (par exemple *COUNT* dans le **tFileRowCount**) dans le champ **Code**, vous devez saisir manuellement le code complet, c'est-à-dire *((Integer)globalMap.get("tFileRowCount_COUNT"))*.
Utilisation	Ce composant est utilisé en tant que composant de milieu de Job, et doit être relié à un composant de début et à un composant de sortie.
Limitation	Il est nécessaire de connaître le langage Java.

Scénario : Transformer des données ligne par ligne avec un tJavaRow

Dans ce scénario, les informations de quelques villes sont lues depuis un fichier d'entrée délimité et sont transformées à l'aide de code Java via le composant **tJavaRow**, puis affichées dans la console.

Construire le Job

1. Déposez un composant **tFileInputDelimited** et un **tJavaRow** de la **Palette** dans l'espace de modélisation graphique et renommez-les afin d'identifier leur rôle dans le Job.

2. Connectez les deux composant à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFileInputDelimited** pour afficher sa vue **Basic settings**.

2. Dans le champ **File name/Stream**, saisissez le chemin d'accès au fichier d'entrée, entre guillemets doubles, ou parcourez votre système en cliquant sur le bouton **[...]** et saisissez *1* dans le champ **Header**.

   Dans cet exemple, le fichier d'entrée contient :

```
City;Population;LandArea;PopDensity
Beijing;10233000;1418;7620
Moscow;10452000;1081;9644
Seoul;10422000;605;17215
Tokyo;8731000;617;14151
New York;8310000;789;10452
```

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir la boîte de dialogue **[Schema]** et définissez la structure des données du fichier d'entrée. Cliquez sur **OK** pour valider le schéma et fermer la boîte de dialogue.

4. Double-cliquez sur le composant **tJavaRow** afin d'afficher sa vue **Basic settings**.

5. Cliquez sur **Sync columns** afin de vous assurer que le schéma est bien récupéré du composant précédent.

6. Dans le champ **Code**, saisissez le code à appliquer à chaque ligne de données.

   Dans cet exemple, vous allez mettre des majuscules aux noms des villes, regrouper les nombres supérieurs à 1000 à l'aide de séparateurs de milliers pour une meilleure lecture et afficher les données dans la console :

```
System.out.print("\n" + input_row.City.toUpperCase() + ":");
System.out.print("\n - Population: "
+ FormatterUtils.format_Number(String.valueOf(input_row.Population), ',', '.') + "
 people");
System.out.print("\n - Land area: "
+ FormatterUtils.format_Number(String.valueOf(input_row.LandArea), ',', '.')
+ " km2");
System.out.print("\n - Population density: "
+ FormatterUtils.format_Number(String.valueOf(input_row.PopDensity), ',', '.') + "
 people/km2\n");
```

Dans le champ **Code**, **input_row** fait référence à la connexion vers le **tJavaRow**.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** dans la vue **Run** pour exécuter le Job.

   Les informations relatives aux villes sont transformées par le code Java défini dans le **tJavaRow** et affichées dans la console.

Execution
Run
Kill
Clear
[statistics] connected
BEIJING:
- Population: 10,233,000 people
- Land area: 1,418 km2
- Population density: 7,620 people/km2
MOSCOW:
- Population: 10,452,000 people
- Land area: 1,081 km2
- Population density: 9,644 people/km2
SEOUL:
- Population: 10,422,000 people
- Land area: 605 km2
- Population density: 17,215 people/km2
TOKYO:
- Population: 8,731,000 people
- Land area: 617 km2
- Population density: 14,151 people/km2
NEW YORK:
- Population: 8,310,000 people
- Land area: 789 km2
- Population density: 10,452 people/km2
[statistics] disconnected
Job dyn_tJavaRow_s1 ended at 17:40 12/04/2012.

tLibraryLoad

Propriétés du tLibraryLoad

Famille de composant	Custom Code	
Fonction	Le **tLibraryLoad** permet d'importer une librairie.	
Objectif	Le **tLibraryLoad** permet de charger des librairies Java utilisables dans un Job.	
Basic settings	*Library*	Sélectionnez dans la liste la librairie à importer, ou cliquez sur le bouton **[...]** afin de parcourir votre répertoire et choisir la librairie que vous souhaitez importer.
Advanced settings	*Dynamic Libs*	**Lib Paths** : Saisissez entre guillemets le chemin d'accès à votre librairie.
	Import	Saisissez le code Java permettant d'importer, si nécessaire, la librairie externe utilisée dans le champ d'édition de code de l'onglet **Basic settings** des composants tels que le **tJava** et le **tJavaFlex**, ou encore le **tJavaMR** dans un Job Map/Reduce.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone, mais il est plus logique de l'utiliser dans un Job.	
Utilisation dans les Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, ce composant est utilisé en standalone. Il génère du code Map/Reduce natif qui peut être exécuté directement dans Hadoop. Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Vérifier le format d'une adresse e-mail

Ce scénario comprend deux composants, un **tLibraryLoad** et un **tJava**. Le but de ce scénario est de vérifier le format d'une adresse e-mail et de retourner si elle est valide ou non.

Construire le Job

1. Dans la **Palette**, ouvrez le dossier **Custom_Code**, puis glissez-déposez les composants **tLibraryLoad** et **tJava** dans l'espace de modélisation graphique.
2. Reliez le **tLibraryLoad** au **tJava** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Configurer le composant tLibraryLoad

1. Double-cliquez sur le **tLibraryLoad** pour afficher sa vue **Basic settings**. Dans la liste **Library**, sélectionnez *jakarta-oro-2.0.8.jar*.
2. Dans le champ **Import** de l'onglet **Advanced settings**, saisissez *import org.apache.oro.text.regex.*;*

Configurer le composant tJava

1. Double-cliquez sur le composant **tJava** pour afficher sa vue **Component**.

2. Dans l'onglet des **Basic settings**, saisissez votre code, comme dans la capture d'écran. Ce code permet de vérifier si une chaîne de caractères correspond à une adresse e-mail, en se basant sur l'expression régulière suivante : `"^[\\w_.-]+@[\\w_.-]+\\.[\\w]+$"`.

Exécuter le Job

Appuyez sur **F6** afin d'enregistrer et d'exécuter votre Job.

La **Console** affiche *false*, c'est-à-dire que l'adresse e-mail n'est pas valide. En effet, son format n'est pas correct.

tSetGlobalVar

Propriétés du tSetGlobalVar

Famille de composant	Custom Code	
Fonction	Le composant **tSetGlobalVar** vous permet de définir et de configurer les variables globales dans l'interface graphique.	
Objectif	Le composant **tSetGlobalVar** simplifie le processus de définition des variables globales.	
Basic settings	*Variables*	Ce tableau contient deux colonnes. **Key** : Nom de la variable à appeler dans le code. **Value** : la valeur assignée à cette variable.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme un sous-job à un composant.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	La connaissance du langage Java est nécessaire.	

Scénario : Afficher le contenu d'une variable globale

Ce scénario se compose d'un Job simple affichant la valeur d'une variable globale définie dans le composant **tSetGlobalVar**.

Construire le Job

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tSetGlobalVar** et **tJava**.

2. Reliez le **tSetGlobalVar** au **tJava** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Configurer le composant tSetGlobalVar

1. Double-cliquez sur le composant **tSetGlobalVar** afin d'afficher sa vue **Basic settings**.

2. Cliquez sur le bouton **[+]** pour ajouter une ligne dans la table **Variables**, puis renseignez les colonnes **Key** et **Value** en saisissant respectivement *K1* et *20*.

3. Double-cliquez ensuite sur le composant **tJava** afin d'afficher sa vue **Basic settings**.

4. Dans la zone **Code**, saisissez le code suivant :

```
String foo = "bar";
String K1;
String Result = "The value is:";
Result = Result + globalMap.get("K1");
System.out.println(Result);
```

Dans ce scénario, utilisez la variable *Result*. Pour accéder à la liste des variables globales, appuyez sur les touches **Ctrl + Espace** de votre clavier et sélectionnez les paramètres correspondants.

Exécuter le Job

Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job SetGlobalVar at 16:22 25/02/2011.

[statistics] connecting to socket on port 3489
[statistics] connected
The value is:20
[statistics] disconnected
Job SetGlobalVar ended at 16:22 25/02/2011.
[exit code=0]
```

Le contenu de la variable globale *K1* s'affiche dans la console.

Composants Data Quality

Ce chapitre passe en revue les propriétés des principaux composants présents dans la famille **Data Quality** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **Data Quality** regroupe les composants vous permettant d'améliorer la qualité de vos données. Ces composants couvrent divers besoin dont la comparaison de colonnes afin d'en extraire des données spécifiques.

tAddCRCRow

Propriétés du tAddCRCRow

Famille de composant	Data Quality	
Fonction	Ce composant calcule une clé de substitution basée sur une ou plusieurs colonne(s) et l'ajoute au schéma défini.	
Objectif	Fournir un ID unique afin d'améliorer la qualité des données traitées.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Implication	Cochez la case correspondant aux colonnes à définir comme clé de substitution et vérifier l'intégrité de leurs données.
Advanced settings	*CRC type*	Sélectionnez la longueur CRC. Plus le CRC est long, moins il y aura de risque d'erreur.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est une étape intermédiaire. Il requiert un flux entrant et une sortie.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Ajouter une clé de substitution à un fichier

Ce scénario décrit un Job ajoutant une clé de substitution à un schéma de fichier délimité.

Construire le Job

1. Glissez les composants suivants dans l'espace de modélisation : **tFileInputDelimited**, **tAddCRCRow** et **tLogRow**.

2. Connectez-les en utilisant un lien de type **Main row**.

Configurer le composant d'entrée

1. Dans la vue **Component** du **tFileInputDelimited**, renseignez le champ **File Name** en saisissant le chemin d'accès à votre fichier, ainsi que toutes les propriétés en rapport, au cas où elles ne soient pas stockées dans le **Repository**.

2. Créez le schéma en cliquant sur le bouton **Edit Schema**, si le schéma n'est pas encore stocké dans le **Repository**. Vérifiez la colonne du type de données, et pour remplir le modèle de date (Date pattern), visitez *http://docs.oracle.com/javase/6/docs/api/index.html* (en anglais).

Configurer le composant tAddCRCRow

1. Dans la vue **Component** du composant **tAddCRCRow**, sélectionnez les cases de la colonne du flux entrant à utiliser pour calculer le CRC.

2. Notez qu'une colonne CRC (en lecture seule) a été ajoutée à la fin du schéma.

3. Dans l'onglet **Advanced settings**, sélectionnez **CRC32** comme **CRC Type** pour avoir une clé de substitution plus longue.

4. Dans la vue **Basic settings** du **tLogRow**, sélectionnez l'option **Table (print values in cells of a table)** pour afficher les données de sortie sous forme de tableau dans la Console.

Exécuter le Job

Sauvegardez votre Job et appuyer sur **F6** pour l'exécuter.

```
Starting job addcrc at 11:51 06/07/2007.
.---------+----------+----------+------+-----------+----------.
|                          tLogRow_1                            |
|=--------+----------+----------+------+-----------+----------=|
|ID_Owners|Reg_Car   |Make      |Color |ID_Reseller|CRC       |
|=--------+----------+----------+------+-----------+----------=|
|1        |1301 DO 05|Citroen   |gold  |38         |27510715125|
|2        |2300 ZP 14|Citroen   |blue  |16         |33211434545|
|3        |4122 JI 74|Renault   |yellow|36         |11525215315|
|4        |3395 QP 05|Citroen   |yellow|51         |14306204562|
|5        |0029 OF 61|Toyota    |red   |37         |10711350076|
|6        |4287 YU 44|Citroen   |blue  |43         |25561510712|
|7        |7119 CQ 97|Honda     |yellow|65         |10136571035|
|8        |3764 PA 47|Renault   |orange|30         |31723253034|
|9        |9939 CJ 88|Mercedes  |red   |41         |27451544441|
|10       |7476 RV 09|Citroen   |grey  |34         |27775721061|
|11       |5287 BP 14|Toyota    |green |27         |2716661270 |
|12       |0750 OG 65|Toyota    |green |8          |23636130023|
|13       |7577 ZQ 59|Volkswagen|purple|55         |37277337005|
```

Une colonne CRC supplémentaire a été ajoutée au schéma, calculée d'après toutes les colonnes précédemment sélectionnées (dans ce cas toutes les colonnes du schéma).

tChangeFileEncoding

Le **tChangeFileEncoding** appartient à deux familles : Data Quality et File. Pour plus d'informations concernant le composant **tChangeFileEncoding**, consultez *tChangeFileEncoding*.

tExtractRegexFields

Le **tExtractRegexFields** appartient à deux familles : Data Quality et Processing. Pour plus d'informations concernant le composant **tExtractRegexFields**, consultez *tExtractRegexFields*.

tFuzzyMatch

Propriétés du tFuzzyMatch

Famille de composant	Data Quality	
Fonction	Le composant **tFuzzyMatch** compare une colonne du flux principal avec une colonne du flux de référence et extrait les données du flux principal en affichant la distance.	
Objectif	Le **tFuzzyMatch** permet d'assurer la qualité des données de n'importe quelle source de données en la comparant avec une source de données de référence.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Deux colonnes en lecture seule sont ajoutées automatiquement au schéma de sortie : **Value** et **Match**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Matching type	Sélectionnez l'algorithme de correspondance approprié parmi les suivants : **Levenshtein** : Basé sur la théorie de la distance d'édition. Il calcule le nombre d'insertion, de suppression ou de substitution nécessaire pour correspondre à la référence. **Metaphone** : Basé sur la phonétique. Tout d'abord, il charge les sonorités de toutes les entrées du flux de référence (Lookup) et les compare avec celles des entrées du flux principal (Main). **Double Metaphone** : une nouvelle version de l'algorithme phonétique Metaphone, qui produit des résultats plus précis que l'algorithme original. Il peut retourner à la fois un code primaire et secondaire pour une chaîne de caractères. Pour éviter toute ambiguïté que le Metaphone n'aurait pas pu détecter, utilisez cette option.
	Min distance	(Levenshtein uniquement) Définissez le nombre minimum de modifications autorisées pour correspondre à la référence. Si vous définissez 0, seules les correspondances exactes sont extraites.
	Max distance	(Levenshtein uniquement) Définissez le nombre maximum de modifications pouvant correspondre à la référence.
	Matching column	Sélectionnez la colonne du flux principal qui doit être comparée avec la colonne clé du flux de référence (Lookup).
	Unique matching	Cochez cette case pour obtenir la meilleure correspondance possible, si plusieurs correspondances sont disponibles.
	Matching item separator	Si plusieurs correspondances sont disponibles, elles seront toutes affichées sauf si la case **Unique match** est cochée. Définissez le délimiteur entre chaque correspondance.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant n'est pas un composant de début (fond vert) et il nécessite deux composants d'entrée et un composant de sortie.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario 1 : Distance de Levenshtein de 0 pour les prénoms

Ce scénario décrit un Job constitué de quatre composants dont le but est de vérifier la distance d'édition comprise entre les colonnes *First Name* (prénom) d'un fichier d'entrée et de comparer les données avec celles du fichier de référence. Le résultat de cette vérification de la distance de Levenshtein et le contenu du flux principal sont affichés dans une table.

Construire le Job

1. Cliquez et déposez les composants suivants de la **Palette** dans l'espace de modélisation : deux **tFileInputDelimited**, un **tFuzzyMatch** et un **tLogRow**.

2. Reliez le premier **tFileInputDelimited** au composant **tFuzzyMatch** à l'aide d'un lien **Row** > **Main**.

3. Connectez le second **tFileInputDelimited** au **tFuzzyMatch** à l'aide d'une connexion de type **Row** > **Main** (qui apparaît comme **Lookup** dans l'espace de modélisation graphique).

4. Reliez le **tFuzzyMatch** au composant de sortie **tLogRow**, à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Configurez les paramètres du premier **tFileInputDelimited** dans sa vue **Basic settings**. Parcourez votre système jusqu'au fichier d'entrée à analyser.

2. Définissez le schéma du composant. Dans cet exemple, le schéma d'entrée possède deux colonnes, *firstname* et *gender*.

3. Configurez le second **tFileInputDelimited** de la même façon.

   Assurez-vous que la colonne de référence a bien été définie comme colonne clé dans le schéma du flux de référence (lookup).

4. Double-cliquez sur le **tFuzzyMatch** pour ouvrir sa vue **Basic settings** et vérifiez son schéma.

   Le schéma doit correspondre à celui du flux d'entrée principal (**Main**) afin que le flux principal soit comparé au flux de référence.

   Notez que deux colonnes : **Value** et **Matching** sont ajoutées au schéma de sortie. Elles correspondent à des informations standard de correspondance et sont en lecture seule.

5. Sélectionnez la méthode à utiliser pour la vérification des données entrantes. Dans ce scénario, la correspondance à sélectionner dans le champ **Matching type** est de type *Levenshtein*.

6. Ensuite définissez la distance. Dans cette méthode, la distance est le nombre de caractères modifiés (insertion, suppression ou substitution) pour que l'entrée principale corresponde exactement à l'entrée de référence.

Dans ce scénario, configurez la distance à 0 au minimum et au maximum dans les champs **Min. distance** et **Max. distance**. Ce qui signifie que seule une correspondance exacte sera redirigée en flux de sortie.

7. Décochez la case **Case sensitive** pour ne pas prendre en compte la casse.

8. Vérifiez que la colonne de correspondance et la colonne de référence sont bien sélectionnées.

9. Laissez les autres paramètres par défaut.

Exécuter le Job

- Enregistrez le Job et appuyez sur la touche **F6** pour exécuter le Job.

Comme la distance d'édition définie est de 0 (min. et max.), la sortie correspond à une jointure entre le flux principal et le flux de référence (lookup), ainsi seules les correspondances totales avec une valeur égale à 0 sont affichées.

Pour un exemple plus évident avec une distance minimum de 1 et une distance maximum de 2, consultez *Scénario 2 : Distance de Levenshtein de 1 ou 2 pour les prénoms*

Scénario 2 : Distance de Levenshtein de 1 ou 2 pour les prénoms

Ce scénario est basé sur le scénario décrit ci-dessus. Seuls les paramètres distance minimum et maximum du composant **tFuzzyMatch** sont modifiés, ce qui modifiera la sortie affichée.

1. Dans la vue **Component** du **tFuzzyMatch**, modifiez la distance minimum de 0 à 1. Ceci exclut directement les correspondances exactes (qui ont une distance de 0).
2. Maintenant mettez une distance maximum de 2. La sortie correspond à toutes les données mises en correspondance contenant une différence de 2 caractères au plus.

   Aucune autre modification de paramètres n'est nécessaire.

3. Assurez-vous que le séparateur de champs (**Matching item separator**) est bien défini, vu que plusieurs entrées de référence peuvent correspondre à l'entrée du flux principal.
4. Enregistrez votre nouveau Job et appuyez sur **F6** pour l'exécuter.

Etant donné que la distance d'édition a été définie à 2, certaines entrées du flux principal correspondent à plusieurs entrées de référence (lookup).

Vous pouvez utiliser une autre méthode, Métaphone, pour évaluer la distance entre le flux principal et le flux de référence, décrite dans le scénario suivant.

Scénario 3 : Distance métaphonique pour les prénoms

Ce scénario est basé sur un scénario décrit plus haut.

1. Modifiez le champ **Matching type** avec l'option **Metaphone**. Aucune distance minimum et aucune distance maximum ne sont à définir puisque cette méthode est basée sur les différences de phonétique avec la référence.

2. Enregistrez le Job et appuyez sur **F6**. La valeur phonétique est affichée avec les correspondances possibles.

tIntervalMatch

Propriétés du tIntervalMatch

Famille de composant	Data Quality	
Fonction	Le composant **tIntervalMatch** permet d'établir une jointure entre un flux principal et un flux de référence. Il classe une valeur spécifiée du flux principal dans la fourchette de valeurs correspondante définie dans le flux de référence et retourne les informations correspondantes.	
Objectif	Il aide à retourner une valeur à l'aide d'une jointure.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Search Column	Sélectionnez la colonne du flux principal contenant les valeurs qui doivent être mises en correspondance avec l'intervalle de valeurs.
	Column (LOOKUP)	Sélectionnez la colonne du flux lookup contenant les valeurs qui doivent être retournées quand la jointure est ok.
	Lookup Column (min) / Include the bound (min)	Sélectionnez la colonne contenant la valeur minimale de l'intervalle. Cochez la case pour inclure la valeur maximale de l'intervalle.
	Lookup Column (max) / Include the bound (max)	Sélectionnez la colonne contenant la valeur maximale de l'intervalle. Cochez la case pour inclure la valeur maximale de l'intervalle.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant manipule un flux de données, il requiert donc un Input et un Output, et nécessite la définition d'une étape intermédiaire.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Identifier l'emplacement de serveurs à partir de leur IP

Ce scénario décrit un Job à quatre composants vérifiant les adresses IP de serveur, listées dans le fichier d'entrée principal, par rapport à une liste d'intervalles d'IPs dans le fichier de référence, afin d'identifier le pays d'hébergement de chaque serveur.

Construire le Job

1. Dans l'espace de modélisation, glissez deux composants **tFileInputDelimited**, un **tIntervalMatch** et un **tLogRow**.

2. Reliez les composants à l'aide de liens **Row** > **Main**.

   La connexion du second **tFileInputDelimited** au **tIntervalMatch** apparaît comme **Lookup**.

Configurer les composants

1. Double-cliquez sur le premier **tFileInputDelimited** pour ouvrir sa vue **Component** et configurer ses propriétés.

2. Parcourez votre système jusqu'au fichier d'entrée principal, contenant une liste des serveurs et de leur adresse IP.

```
Server;IP
Server1;057.010.010.010
Server2;001.010.010.100
Server3;057.030.030.030
Server4;053.010.010.100
```

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir la boîte de dialogue **[Schema]** et configurer le schéma d'entrée. Selon la structure du fichier d'entrée, le schéma se compose de deux colonnes, *Server* et *IP*, de type **String**. Cliquez sur **OK** pour fermer la boîte de dialogue.

4. Dans le champ **Header**, saisissez le nombre de lignes d'en-tête à ignorer et laissez les autres champs tels qu'ils sont.

5. Configurez de la même manière les propriétés du second **tFileInputDelimited**.

Le fichier à utiliser en entrée pour le flux de référence liste, dans cet exemple, des intervalles d'adresses IP et le pays correspondant.

```
StartIP;EndIP;Country
001.000.000.000;001.255.255.255;USA
002.006.190.056;002.006.190.063;UK
011.000.000.000;011.255.255.255;USA
057.000.000.000;057.255.255.255;France
012.063.178.060;012.063.178.063;Canada
053.000.000.000;053.255.255.255;Germany
```

Le schéma du flux de référence doit posséder la structure suivante :

6. Double-cliquez sur le composant **tIntervalMatch** pour ouvrir sa vue **Basic settings**.

7. Dans la liste **Search Column**, sélectionnez le flux d'entrée principal contenant les valeurs à mettre en correspondance avec les intervalles de valeurs. Dans cet exemple, les adresses IP des serveurs doivent correspondre aux intervalles de valeurs du flux de référence.

8. Dans la liste **Column (LOOKUP)**, sélectionnez la colonne de référence contenant les valeurs à retourner. Dans cet exemple, il s'agit du nom des pays où se trouvent les serveurs.

9. Configurez les colonnes de référence minimale et maximale correspondant aux limites de l'intervalle défini dans le schéma de référence, *StartIP* et *EndIP*, respectivement.

Exécuter le Job

- Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job et appuyez sur **F6** pour l'exécuter.

  Le nom du pays où est hébergé chaque serveur s'affiche à côté de l'adresse IP correspondante, dans la console.

tReplaceList

Propriétés du tReplaceList

Famille de composant	Data Quality	
Fonction	Le **tReplaceList** effectue un rechercher/remplacer dans les colonnes d'entrée spécifiées en fonction des données de référence (lookup).	
Objectif	Ce composant permet de nettoyer tous les fichiers avant un traitement futur.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Deux colonnes en lecture seule, **Value** et **Match** sont automatiquement ajoutés au schéma de sortie. *Le type de données défini dans les schémas doit être cohérent, c'est-à-dire qu'un entier ne peut être remplacé que par un entier en utilisant un entier comme champ de référence. Une valeur d'un type ne peut être remplacée par une valeur d'un autre type.*
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Lookup search column	Sélectionnez la colonne à rechercher dans le schéma de lookup.
	Lookup replacement column	Sélectionnez la colonne dans laquelle les valeurs de remplacement sont stockées.
	Column options	Sélectionnez les colonnes du flux principal dans lequel le remplacement est effectué.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire, il nécessite un composant d'entrée et un composant de sortie.	

Scénario : Remplacement à partir d'un fichier de référence

Le Job suivant recherche et remplace une liste d'états par le code à deux lettres correspondant. Les codes correspondants proviennent d'un fichier de référence placé dans le flux lookup du Job.

Construire le Job

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : deux **tFileInputDelimited**, un **tReplaceList** et un **tLogRow**.

2. Reliez les deux **tFileInputDelimited** au **tReplaceList** à l'aide de liens **Row** > **Main**. Notez que le lien entre le composant de référence (le second **tFileInputDelimited**) et le **tReplaceList** apparaît comme lien **Lookup**.

3. Reliez le composant **tReplaceList** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le premier **tFileInputDelimited** pour ouvrir sa vue **Basic settings** et configurer les paramètres du flux d'entrée principal, notamment le chemin d'accès au fichier à lire, ainsi que le nombre de lignes d'en-tête à ignorer.

Dans cet exemple, le fichier principal d'entrée fournit une liste de personnes et d'états des Etats-Unis. Voici un extrait du fichier :

```
name;state
Andrew Kennedy;Mississippi
Benjamin Carter;Louisiana
Benjamin Monroe;West Virginia
Bill Harrison;Tennessee
Calvin Grant;Virginia
Chester Harrison;Rhode Island
Chester Hoover;Kansas
Chester Kennedy;Maryland
Chester Polk;Indiana
Dwight Nixon;Nevada
Dwight Roosevelt;Mississippi
Franklin Grant;Nebraska
```

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir la boîte de dialogue **[Schema]** et configurer le schéma d'entrée.

   Selon la structure du fichier d'entrée principal, le schéma d'entrée doit contenir deux colonnes : *name* et *state*.

   Cliquez sur **OK** pour fermer la boîte de dialogue et propager les modifications au composant suivant.

3. Configurez de la même façon les propriétés du second **tFileInputDelimited**.

Dans cet exemple, le fichier de référence fournit une liste d'états ainsi que leur code à deux lettres. Le schéma de référence doit donc comprendre deux colonnes : *state* et *code*.

4. Double-cliquez sur le **tReplaceList** pour ouvrir sa vue **Basic settings** et paramétrer l'opération à effectuer.

5. Dans la liste **Lookup search column**, sélectionnez la colonne à rechercher. Dans ce scénario, recherchez la colonne *state*.

6. Dans la liste **Lookup replacement column**, sélectionnez la colonne contenant les valeurs de remplacement, *code* pour les codes à deux lettres des états, dans cet exemple.

7. Dans la table **Column options**, cochez la case **Replace** pour la colonne *states*, afin de remplacer le nom des états par le code correspondant.

8. Dans le composant **tLogRow**, sélectionnez l'option **Table** pour une meilleure lisibilité de la sortie.

Exécuter le Job

- Sauvegardez le Job et appuyez sur **F6** pour l'exécuter.

Le nom des états a été remplacé par le code à deux lettres correspondant.

tSchemaComplianceCheck

Propriétés du tSchemaComplianceCheck

Famille de composant	Data Quality	
Fonction	Ce composant valide toutes les lignes d'entrée en fonction du schéma de référence ou vérifie les types, la nullabilité, la longueur de la ligne en fonction des valeurs de référence. Cette validation peut être appliquée à l'ensemble des lignes ou une partie seulement.	
Objectif	Le **tSchemaComplianceCheck** permet d'assurer la qualité des données de n'importe quelle source de données en les comparant aux données de référence.	
Basic settings	*Base Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**) soit distant dans le **Repository**. Définissez la structure et la nature des données à traiter. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Check all columns from schema	Cochez cette option si vous souhaitez effectuer la vérification sur toutes les colonnes du schéma de base.
	Custom defined	Cochez cette option si vous souhaitez effectuer la vérification sur des colonnes en particulier. Lorsque cette option est cochée, le tableau **Checked Columns** apparaît.
	Checked Columns	Définissez dans ce tableau quelles sont les vérifications à effectuer, et sur quelles colonnes.
		Column : Affiche les noms des colonnes.
		Type : sélectionnez les propriétés que les données doivent avoir. Cette validation est obligatoire pour toutes les colonnes. Pour effectuer cette vérification sur toutes les colonnes, cochez la case d'en-tête du tableau.
		Date pattern : Définissez le format de date souhaité sur chaque colonne avec le type de données *Date*.
		Nullable : Cochez la case de la colonne que vous souhaitez définir comme colonne nullable, afin de permettre aux lignes vide dans cette colonne de se diriger vers le flux de sortie sans tenir compte du schéma de base défini. Pour définir toutes les colonnes comme nullables, cochez la case d'en-tête du tableau.
		Undefined or empty : Cochez la case de la colonne dont vous souhaitez rejeter les lignes vides, alors que la colonne n'est pas nullable dans la définition du schéma de base. Pour effectuer cette vérification sur toutes les colonnes, cochez la case au niveau de l'en-tête de la table.
		Max length : Cochez la case de la colonne dont vous souhaitez vérifier la longueur des données en fonction de la longueur définie dans le schéma de base. Pour effectuer cette vérification sur toutes les colonnes, cochez la case d'en-tête du tableau.
	Use another schema for compliance check	Définissez un schéma de référence en donnant aux données les propriétés qu'elles sont censées avoir. Ainsi, les données non conformes seront rejetées.

		Ces propriétés peuvent correspondre au type de données, aux valeurs nulles, et/ou à la longueur des données.
	Trim the excess content of column when length checking chosen and the length is greater than defined length	Cochez cette case afin de tronquer les données dépassant la longueur spécifiée plutôt que de les rejeter. L'option fonctionne avec les trois modes du **tSchemaComplianceCheck**. Cette option s'applique uniquement aux données de type String.
Advanced settings	*Use Fastest Date Check*	Cochez cette case afin d'effectuer une vérification rapide du format de date en utilisant la méthode *TalendDate.isDate()* de la routine système *TalendDate* lorsque *Date pattern* (format de date) n'est pas défini. Pour plus d'informations concernant les routines, consultez le *Guide utilisateur du Studio Talend*.
	Ignore TimeZone when Check Date	Cochez cette case pour ignorer le fuseau horaire lors de la vérification de la date. Cette case n'est pas disponible lorsque le mode **Check all columns from schema** est sélectionné.
	Treat all empty string as NULL	Cochez cette case afin de traiter tous les champs vides de chaque colonne comme des valeurs nulles, plutôt que comme des chaînes de caractères vides. Cette case est cochée par défaut. Lorsqu'elle est décochée, le tableau **Choose Column(s)** s'affiche pour vous permettre de sélectionner les colonnes que vous souhaitez.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n'est pas disponible dans la version Map/Reduce de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire permettant d'exclure les données non conformes du flux principal. Ce composant ne peut être utilisé comme composant de début puisqu'il requiert un flux d'entrée. Il nécessite aussi au moins un composant de sortie afin de regrouper les flux validés et éventuellement un deuxième pour les données rejetées via un lien **Rejects**. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Le composant ne supporte pas les données de type Object et List. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	

Scénario : Valider les dates en fonction d'un schéma

Ce scénario permet de vérifier le type, la nullabilité et la longueur des données du flux d'entrée en fonction du schéma de référence défini. Il permet également d'afficher les résultats de validation dans la console de la vue **Run**.

Le flux d'entrée provient d'un fichier csv simple contenant des données dont les propriétés ont un mauvais type, un mauvais id, dont les données sont supérieures à la longueur maximum, ou contenant des valeurs nulles dans les colonnes alors que celles-ci sont définies comme non nullables.

```
ID;Name;BirthDate;State;City
1;Dwight;06-04-2008;Delaware;Concord
2;Warren;25-10-2008;Montana
3;Benjamin;17-08-2008;Washington;Austin
4;Harry;14-04-2008;Kansas;Annapolis
5;Ulysses;2007-04-12;Michigan;Raleigh
6;James;19-08-2007;Delaware;Charleston
.7;Bill;20-04-2007;Illinois;Bismarck
8;Ulysses;04-12-2008;;Saint Paul
9;Thomas;09-05-2008;Maryland;Albany
10;Ronald;11-02-2008;Florida;Hartford
```

Construire le Job

1. A partir de la **Palette**, cliquez et déposez les composants suivants dans l'espace de modélisation graphique : **tFileInputDelimited**, **tSchemaComplianceCheck**, ainsi que deux composants **tLogRow**.
2. Cliquez-droit sur le **tFileInputDelimited** pour le connecter au composant **tSchemaComplianceCheck** via une connexion **Row Main**.
3. Cliquez-droit sur le composant **tSchemaComplianceCheck** et sélectionnez **Row > Main** pour le connecter au premier composant **tLogRow**. Ce flux de sortie ne contiendra que les données valides.
4. Cliquez-droit une deuxième fois sur le composant **tSchemaComplianceCheck** et cette fois sélectionnez **Row > Rejects** pour le connecter au second composant **tLogRow**. Cette deuxième sortie regroupera toutes les données non conformes. Il passe deux colonnes supplémentaires au composant suivant : *ErrorCode* et *ErrorMessage*. Ces deux colonnes en lecture seule fournissent des informations concernant les données rejetées afin de simplifier la gestion d'erreurs et la résolution de problèmes, si nécessaire.

Configurer les composants

1. Double-cliquez sur le composant **tFileInputDelimited** afin d'afficher sa vue **Basic settings**. Configurez ses propriétés, notamment le nom du fichier d'entrée et le nombre de lignes d'en-tête à ignorer.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit Schema** pour décrire la structure de données de ce fichier d'entrée. Dans ce scénario, le schéma est composé de cinq colonnes : *ID*, *Name*, *BirthDate*, *State* et *City*.

3. Renseignez le champ **Length** des colonnes *Name*, *State* et *City* : respectivement *7*, *10* et *10*, puis cliquez sur **OK** pour fermer la boîte de dialogue et propager le schéma.

4. Double-cliquez sur le composant **tSchemaComplianceCheck** pour configurer les paramètres de validation.

5. Cochez l'option **Custom defined** dans la zone **Mode** pour effectuer des validations personnalisées. Dans cet exemple, le tableau **Checked columns** permet de configurer les paramètres de validations.

   Cependant, vous pouvez aussi cocher la case **Check all columns from schema** si vous souhaitez vérifier tous les paramètres de toutes les colonnes (type, valeurs nulles et longueur), en fonction du schéma de base défini, ou encore cocher l'option **Use another schema for compliance check** et définir le schéma de la structure des données souhaité.

6. Dans le tableau **Checked Columns**, définissez les vérifications à effectuer. Pour ce scénario :

   - Sélectionnez **Int** comme **Type** de données de la colonne *ID*.

   - Cochez la case **Length** afin que les longueurs des colonnes *Name*, *State* et *City* soient vérifiées.

   - Sélectionnez **Date** comme **Type** de données de la colonne *BirthDate* et *dd-MM-yyyy* comme **Date pattern**.

   - Décochez la case **Nullable** afin que les valeurs nulles de toutes les colonnes soient vérifiées.

   Pour rejeter les champs dont les données sont supérieures à la longueur maximum définie, assurez-vous que la case **Trim the excess content of column when length checking chosen and the length is greater than defined length** est décochée.

7. Dans la vue **Advanced settings** du composant **tSchemaComplianceCheck**, cochez la case **Treat all empty string as NULL** afin de rejeter toutes les lignes contenant des champs vides.

8. Pour visualiser le résultat du processus de validation sous forme de tableau dans la console **Run**, double-cliquez sur chaque composant **tLogRow** et cochez la case **Table** dans leur vue **Basic settings**.

Exécuter le Job

- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
[statistics] connected
.--+--------+----------+----------+----------+---------+-----------------------------------------.
|                                          tLogRow_2                                              |
|=-+--------+----------+----------+----------+---------+-----------------------------------------=|
|ID|Name    |BirthDate |State     |City      |errorCode|errorMessage                             |
|=-+--------+----------+----------+----------+---------+-----------------------------------------=|
|2 |Warren  |25-10-2008|Montana   |          |4        |City:empty or null                       |
|3 |Benjamin|17-08-2008|Washington|Austin    |8        |Name:exceed max length                   |
|5 |Ulysses |2007-04-12|Michigan  |Raleigh   |2        |BirthDate:wrong DATE pattern or wrong DATE data|
|.7|Bill    |20-04-2007|Illinois  |Bismarck  |2        |ID:wrong type                            |
|8 |Ulysses |04-12-2008|          |Saint Paul|4        |State:empty or null                      |
'--+--------+----------+----------+----------+---------+-----------------------------------------'
.--+------+----------+--------+----------.
|               tLogRow_1                |
|=-+------+----------+--------+---------=|
|ID|Name  |BirthDate |State   |City      |
|=-+------+----------+--------+---------=|
|1 |Dwight|06-04-2008|Delaware|Concord   |
|4 |Harry |14-04-2008|Kansas  |Annapolis |
|6 |James |19-08-2007|Delaware|Charleston|
|9 |Thomas|09-05-2008|Maryland|Albany    |
|10|Ronald|11-02-2008|Florida |Hartford  |
'--+------+----------+--------+----------'

[statistics] disconnected
```

Deux tableaux s'affichent dans la console, qui montrent respectivement les données valides et les données rejetées.

tUniqRow

Propriétés du tUniqRow

Famille de composant	Data Quality	
Fonction	Le composant **tUniqRow** compare les entrées et supprime les doublons du flux d'entrée.	
Objectif	Ce composant assure une qualité de données des flux d'entrée et de sortie du Job.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Unique key	Sélectionnez dans cette zone une ou plusieurs colonnes sur lesquelles le dédoublonnage sera effectué. - Cochez la case **Key attribute** afin d'effectuer le dédoublonnage sur toutes les colonnes. - Cochez la case **Case sensitive** afin de différencier les majuscules et les minuscules.
Advanced settings	*Only once each duplicated key*	Cochez cette case si vous souhaitez envoyer uniquement les premières entrées en doublon des colonnes définies comme clé(s) vers le flux de sortie des doublons.
	Use of disk (suitable for processing large row set) Indisponible pour les Jobs Map/Reduce.	Cochez cette case pour permettre la génération de fichiers temporaires sur le disque dur lors du traitement de données volumineuses. Cela permet d'empêcher l'échec de l'exécution d'un Job dû à un débordement de la mémoire. Quand cette case est cochée, vous devez également définir les éléments suivants : - **Buffer size in memory**: Sélectionnez le nombre de lignes à mettre en mémoire tampon avant qu'un fichier temporaire ne soit généré sur le disque dur. - **Directory for temp files**: Indiquez l'endroit où les fichiers temporaires doivent être enregistrés.

		Assurez-vous que le répertoire indiqué pour vos fichiers temporaires existe, sans quoi l'exécution du Job échouera.
	Ignore trailing zeros for BigDecimal	Cochez cette case pour ignorer les zéros en fin de champs pour les données de type BigDecimal.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. Notez que cette case n'est pas disponible dans la version Map/Reduce de ce composant.
Global Variables	**NB_UNIQUES** : nombre de lignes uniques. Cette variable est une variable **After** et retourne un nombre entier. **NB_DUPLICATES** : nombre de lignes en doublon. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire puisqu'il sert à traiter un flux de données. Ainsi, il nécessite un composant d'entrée et un composant de sortie.	
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Pour un exemple de scénario de Job Map/Reduce Job utilisant ce composant, consultez le *Scénario 2 : Dédoublonner des entrées en utilisant des composants Map/Reduce*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Dédoublonner des données

Ce scénario décrit un Job à cinq composants, dont le but est de trier des entrées à partir d'une liste d'entrée comprenant des noms, de trouver des noms en doublons, et d'afficher les noms uniques ainsi que les noms en doublon dans la console **Run**.

Construire le Job

1. A partir de la **Palette**, cliquez et déposez les composants suivants dans l'espace de modélisation graphique : un **tFileInputDelimited**, un **tSortRow**, un **tUniqRow**, ainsi que deux composants **tLogRow**, et nommez-les comme illustré ci-dessus.
2. Connectez les composants **tFileInputDelimited**, **tSortRow**, et **tUniqRow** à l'aide de liens **Row** > **Main**.
3. Connectez le **tUniqRow** au premier **tLogRow** à l'aide d'un lien **Main** > **Uniques**.
4. Connectez le composant **tUniqRow** au second **tLogRow** via une connexion de type **Main** > **Duplicates**.

Configurer les composants

1. Double-cliquez sur le composant **tFileInputDelimited** afin d'afficher l'onglet **Basic settings** de sa vue **Component**.

2. Cliquez sur le bouton **[...]** à côté du champ **File Name** pour sélectionner le fichier contenant les données d'entrée.
3. Définissez l'en-tête (**Header**) et le pied de page (**Footer**). Dans ce scénario, la ligne d'en-tête est la première ligne du fichier d'entrée.
4. Cliquez sur **Edit schema** pour définir le schéma de ce composant. Dans ce scénario, le fichier d'entrée comprend cinq colonnes : *Id*, *FirstName*, *LastName*, *Age*, et *City*. Cliquez ensuite sur **OK** pour propager le schéma et fermer l'éditeur de schéma.
5. Double-cliquez sur le **tSortRow** afin d'afficher sa vue **Basic settings**.

6. Afin de disposer les entrées en fonction de l'ordre alphabétique des noms, ajoutez deux lignes au tableau **Criteria** en cliquant sur le bouton **[+]**, sélectionnez les colonnes *FirstName* et *LastName* sous **Schema column**, sélectionnez le type de tri alphabétique (*alpha)*, puis sélectionnez l'ordre ascendant (asc).

7. Double-cliquez sur le composant **tUniqRow** afin d'afficher l'onglet **Basic settings** de sa vue **Component**.

8. Dans la zone **Unique key**, sélectionnez les colonnes sur lesquelles vous souhaitez effectuer le dédoublonnage. Dans ce scénario, les noms en doublon seront triés.

9. Dans l'onglet **Basic settings** de la vue **Component** des deux composants **tLogRow**, cochez l'option **Table** afin de visualiser le résultat de l'exécution du Job en mode tableau.

Sauvegarder et exécuter le Job

1. Sauvegardez votre Job à l'aide des touches **Ctrl+S**.

2. Exécutez le Job en appuyant sur la touche **F6** ou en cliquant sur le bouton **Run** de l'onglet **Run**.

   Dans la console **Run**, les noms uniques et les noms en doublon sont affichés dans des tableaux différents.

```
|70 |Rutherford|Garfield  |74 |Honolulu       |
|1  |Rutherford|Roosevelt |56 |Oklahoma City  |
|57 |Theodore  |Tyler     |42 |Bismarck       |
|84 |Thomas    |Eisenhower|65 |Concord        |
|9  |Thomas    |Lincoln   |44 |Augusta        |
|15 |Ulysses   |Johnson   |37 |Austin         |
|39 |Ulysses   |Kennedy   |27 |Phoenix        |
|80 |Ulysses   |Pierce    |56 |Madison        |
|25 |Warren    |Jefferson |40 |Frankfort      |
|87 |Warren    |Johnson   |66 |Oklahoma City  |
|36 |William   |Carter    |36 |Trenton        |
|18 |William   |Hoover    |76 |Columbus       |
|59 |Woodrow   |Fillmore  |42 |Saint Paul     |
|78 |Woodrow   |Roosevelt |66 |Atlanta        |
|38 |Woodrow   |Taft      |35 |Oklahoma City  |
|98 |Woodrow   |Tyler     |75 |Concord        |
|16 |Zachary   |Monroe    |35 |Richmond       |
|11 |Zachary   |Taft      |63 |Helena         |
`---+----------+----------+---+---------------'

.--+---------+----------+---+---------.
|            Duplicated_Names          |
|=-+---------+----------+---+---------=|
|Id|FirstName|LastName  |Age|City     |
|=-+---------+----------+---+---------=|
|22|Calvin   |Eisenhower|27 |Montgomery|
|79|Calvin   |Eisenhower|67 |Juneau   |
|24|George   |Harrison  |74 |Harrisburg|
|69|Ulysses  |Johnson   |71 |Olympia  |
`--+---------+----------+---+---------'
```

Scénario 2 : Dédoublonner des entrées en utilisant des composants Map/Reduce

Ce scénario montre comment créer un Job **Talend** Map/Reduce afin de dédoublonner des entrées. En d'autres termes, ce scénario utilise des composants Map/Reduce afin de générer du code Map/Reduce et exécuter le Job dans Hadoop.

Notez que les composants **Talend** Map/Reduce ne sont disponibles que si vous avez souscrit à l'une des solutions Big Data de **Talend** et que ce scénario ne peut être reproduit qu'avec des composants Map/Reduce.

Les données d'exemple utilisées dans ce scénario se présentent comme suit :

```
1;Harry;Ford;68;Albany
2;Franklin;Wilson;79;Juneau
3;Ulysses;Roosevelt;25;Harrisburg
4;Harry;Ford;48;Olympia
5;Martin;Reagan;75;Columbia
6;Woodrow;Roosevelt;63;Harrisburg
7;Grover;McKinley;98;Atlanta
8;John;Taft;93;Montpelier
9;Herbert;Johnson;85;Lincoln
10;Grover;McKinley;33;Lansing
```

Etant donné que le *Studio Talend* vous permet de convertir un Job **Map/Reduce** en Job **Standard** (non Map/Reduce) et vice-versa, vous pouvez convertir le scénario présenté précédemment afin de créer ce Job Map/Reduce. Ainsi, la plupart des composants utilisés peuvent garder leurs paramètres d'origine afin de réduire votre charge de travail pour la création de ce Job.

Avant de commencer ce scénario, assurez-vous de disposer des droits d'accès Hadoop nécessaires pour accéder à la distribution Hadoop utilisée. Procédez ensuite comme suit :

Convertir le Job

1. Dans le **Repository** de la perspective **Integration** du *Studio Talend*, cliquez-droit sur le Job que vous avez créé dans le scénario précédent afin d'ouvrir son menu contextuel. Sélectionnez ensuite **Edit properties**.

   L'assistant **[Edit properties]** s'ouvre alors. Notez que le Job doit être fermé avant que vous ne puissiez effectuer des changements dans cet assistant.

   L'assistant se présente comme dans la capture d'écran ci-dessous :

Notez que vous pouvez changer le nom du Job ainsi que les autres informations de description du Job à partir de cet assistant.

2. Cliquez sur **Convert to Map/Reduce Job**. Un Job Map/Reduce portant le même nom apparaît sous le sous-nœud **Map/Reduce Jobs** du nœud **Job Design**.

Si vous devez partir d'un Job Map/Reduce Job vierge, cliquez-droit sur le nœud **Job Design** ou sur le sous-nœud **Map/Reduce Jobs** et sélectionnez **Create Map/Reduce Job** dans le menu contextuel. Un Job vierge s'ouvre dans l'espace de modélisation graphique. Pour plus d'informations, consultez la section décrivant comment créer un Job Map/Reduce dans le *Guide de prise en main de Talend Open Studio for Big Data*.

Réorganiser les composants

1. Double-cliquez sur le Job Map/Reduce que vous venez de créer afin de l'ouvrir dans l'espace de modélisation graphique. La **Palette** de composants Map/Reduce s'ouvre et des composants peuvent être barrés, car il n'ont pas de version Map/Reduce.

2. Cliquez-droit sur chacun des composants en question et sélectionnez **Delete** afin de les supprimer.

3. Déposez un composant **tHDFSInput**, un **tHDFSOutput** et un **tJDBCOutput** dans l'espace de modélisation graphique. Le **tHDFSInput** lit les données à partir de la distribution Hadoop utilisée, le **tHDFSOutput** écrit les données dans cette distribution et le **tJDBCOutput** écrit les données dans une base de données, par exemple, une base de données MySQL pour ce scénario. Les deux composants de sortie remplacent les deux **tLogRow** pour écrire les données en sortie.

Si vous partez d'un Job vierge, déposez également un **tSortRow** et un **tUniqRow**.

4. Connectez le **tHDFSInput** au **tSortRow** à l'aide d'un lien **Row > Main** et acceptez le schéma provenant du **tSortRow**.

5. Connectez le **tUniqRow** au **tHDFSOutput** à l'aide d'un lien **Row > Uniques** et au **tJDBCOutput** à l'aide d'un lien **Row > Duplicates**.

Configurer une connexion Hadoop

1. Cliquez sur **Run** afin d'ouvrir la vue correspondante et cliquez sur l'onglet **Hadoop Configuration** afin d'afficher cette vue et configurer la connexion Hadoop pour ce Job.

   La vue se présente comme dans la capture d'écran ci-dessous :

2. Dans la liste **Property type**, sélectionnez **Built-in**. Si vous avez créé une connexion afin qu'elle soit utilisable depuis le **Repository**, sélectionnez alors **Repository** afin que le *Studio Talend* réutilise les informations de connexion pour ce Job.

   Pour plus d'informations concernant la création d'une connexion Hadoop dans le **Repository**, consultez le chapitre décrivant le nœud **Hadoop cluster** du *Guide de prise en main de Talend Open Studio for Big Data*.

3. Dans la zone **Version**, sélectionnez la distribution Hadoop à utiliser et sa version. Si vous ne trouvez pas la distribution correspondant à la votre dans la liste, sélectionnez **Custom** afin de vous connecter à une distribution Hadoop non officiellement supportée par le *Studio Talend*.

Pour un exemple d'utilisation étape par étape de cette option **Custom**, consultez *Connexion à une distribution Hadoop personnalisée*.

En plus de l'évolution de Hadoop, veuillez noter les changements suivants :

- Si vous utilisez **Hortonworks Data Platform V2.2**, les fichiers de configuration de votre cluster peuvent utiliser des variables d'environnement comme *${hdp.version}*. Si vous êtes dans cette situation, vous devez configurer la propriété *mapreduce.application.framework.path* dans la table **Hadoop properties**, avec la valeur du chemin d'accès pointant explicitement vers l'archive du framework MapReduce de votre cluster. Par exemple :

```
mapreduce.application.framework.path=/hdp/apps/2.2.0.0-2041/mapreduce/
mapreduce.tar.gz#mr-framework
```

- Si vous utilisez **Hortonworks Data Platform V2.0.0**, le type de système d'exploitation utilisé pour la distribution Hadoop et pour le Job **Talend** doit être identique, par exemple Windows ou Linux. Sinon, vous devez utiliser le Jobserver de **Talend** afin d'exécuter un Job dans le même type de système d'exploitation que celui dans lequel la distribution **Hortonworks Data Platform V2.0.0** utilisée est exécutée. Pour plus d'informations concernant le Jobserver de **Talend**, consultez le *Guide d'installation et de migration Talend*.

4. Dans le champ **Name node**, saisissez l'emplacement du nœud maître, le NameNode, de la distribution utilisée. Par exemple, *hdfs://tal-qa113.talend.lan:8020*.

   Si vous utilisez une distribution MapR, vous pouvez simplement laisser *maprfs:///* dans ce champ. Le client MapR se charge de créer la connexion au moment de l'exécution. Notez que le client doit être correctement installé. Pour plus d'informations concernant la configuration d'un client MapR, consultez la page suivante sur le site de la documentation de MapR : http://doc.mapr.com/display/MapR/Setting+Up+the+Client (en anglais).

5. Dans le champ **Job tracker**, saisissez l'emplacement du JobTracker de votre distribution. Par exemple, *tal-qa114.talend.lan:8050*.

   Notez que le mot "Job" dans le terme "JobTracker" désigne les Jobs MR ou Map/Reduce décrits dans la documentation Apache disponible sur le site Internet d'Apache http://hadoop.apache.org/ (en anglais).

   Si vous utilisez YARN dans votre cluster Hadoop, comme **Hortonworks Data Platform V2.0.0** ou **Cloudera CDH4.3 + (mode YARN)**, vous devez spécifier l'emplacement du **Resource Manager** et non celui du Jobtracker. Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser :

   - Cochez la case **Set resourcemanager scheduler address** et saisissez l'adresse de l'ordonnanceur (Scheduler) dans le champ qui apparaît.

   - Cochez la case **Set jobhistory address** et saisissez l'emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory.

   - Cochez la case **Set staging directory** et saisissez le chemin d'accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l'exécution de programmes. Ce répertoire se trouve sous la propriété *yarn.app.mapreduce.am.staging-dir* dans les fichiers de configuration, notamment les fichiers *yarn-site.xml* et *mapred-site.xml* de votre distribution.

   - Cochez la case **Use datanode hostname** pour permettre au Job d'accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété *dfs.client.use.datanode.hostname* à *true*. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.

6. Si la distribution utilisée nécessite une authentification Kerberos, cochez la case **Use Kerberos authentication** et saisissez les détails d'authentification. Sinon, laissez cette case décochée.

Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d'utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

De plus, comme ce composant effectue des calculs Map/Reduce, vous devez également authentifier les services associés, comme le serveur de l'historique des Jobs et le gestionnaire de ressources ou le JobTracker, selon votre distribution, dans le champ correspondant. Ces principaux se trouvent dans les fichiers de configuration de votre distribution. Par exemple, dans une distribution CDH4, le principal du gestionnaire de ressource est configuré dans le fichier *yarn-site.xml* et le principal de l'historique des Job dans le fichier *mapred-site.xml*.

Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case **Use a keytab to authenticate**. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**.

Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.

7. Dans le champ **User name**, saisissez votre nom d'utilisateur pour votre distribution. Si vous laissez ce champ vierge, le nom d'utilisateur de la machine hébergeant le *Studio Talend* est utilisé.

8. Dans le champ **Temp folder**, saisissez le chemin d'accès au dossier HDFS dans lequel sont stockés les fichiers temporaires générés pendant les opérations de Map/Reduce.

9. Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d'autres termes, changez le séparateur si celui-ci n'est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.

10. Laissez la case **Clear temporary folder** (Effacer le contenu du dossier temporaire) cochée, sauf si vous souhaitez garder ces fichiers temporaires.

11. Laissez la case **Compress intermediate map output to reduce network traffic** cochée, afin de réduire le temps de transfert des partitions du Mapper aux différents Reducers.

    Cependant, si le transfert des données dans le Job est négligeable, il est recommandé de décocher cette case afin de désactiver l'étape de compression, car la compression consomme des ressources supplémentaires du CPU.

12. Si vous souhaitez utiliser des propriétés Hadoop personnalisées, renseignez la table **Hadoop properties** avec la ou les propriété(s) à personnaliser. Lors de l'exécution, ces modifications écrasent les propriétés correspondantes par défaut utilisées par le Studio pour son moteur Hadoop.

    Pour plus d'informations concernant les propriétés requises par Hadoop, consultez la documentation Apache Hadoop sur le site http://hadoop.apache.org (en anglais), ou la documentation de la distribution Hadoop que vous utilisez.

13. Si le système Hadoop utilisé est Hortonworks Data Platform V1.2 ou Hortonworks Data Platform V1.3, vous devez définir des allocations de mémoire suffisantes pour que les opérations de map et reduce soient effectuées par le système Hadoop.

    Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations *map* et *reduce* dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux *1000*, ce qui est normalement adapté pour l'exécution de ces opérations.

Si la distribution utilisée est YARN, les paramètres de mémoire à définir sont alors **Map (in Mb)**, **Reduce (in Mb)** et **ApplicationMaster (in Mb)**, respectivement. Ces champs permettent d'allouer dynamiquement de la mémoire aux opérations map et reduce et à l'ApplicationMaster de YARN.

Pour plus d'informations sur l'onglet **Hadoop Configuration**, consultez la section décrivant comment configurer la connexion Hadoop pour un Job **Talend** Map/Reduce dans le *Guide de prise en main de Talend Open Studio for Big Data*.

Pour plus d'informations sur le Resource Manager, son ordonnanceur et l'ApplicationMaster, consultez la documentation de YARN sur le site http://hortonworks.com/blog/apache-hadoop-yarn-concepts-and-applications/ (en anglais).

Pour plus d'informations concernant la définition des paramètres de mémoire pour YARN et MapReduce, consultez la documentation de la distribution utilisée comme, par exemple, celle d'Hortonworks : http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.0.6.0/bk_installing_manually_book/content/rpm-chap1-11.html (en anglais).

Configurer les composants d'entrée et de sortie

Configurer le tHDFSInput

1. Double-cliquez sur le **tHDFSInput** pour ouvrir sa vue **Component**.

2. Cliquez sur le bouton ... situé à côté du champ **Edit schema** afin de vérifier si le schéma reçu dans les étapes précédentes a été correctement défini.

Notez que si vous avez créé ce Job à partir d'un Job vierge, vous devez cliquer sur le bouton afin d'ajouter manuellement ces colonnes au schéma. Si le schéma a été défini dans le **Repository**, vous pouvez sélectionner l'option **Repository** dans la liste **Schema** de la vue **Basic settings** afin de le réutiliser. Pour plus d'informations sur comment définir un schéma dans le **Repository**, consultez le chapitre décrivant la gestion des métadonnées dans le *Guide utilisateur de*Studio Talend ou le chapitre décrivant le nœud **Hadoop cluster** du **Repository** dans le *Guide de prise en main de Talend Open Studio for Big Data*.

3. Si vous effectuez des changements dans le schéma, cliquez sur **OK** afin de valider ces changements et acceptez la propagation proposée par la boîte de dialogue qui s'affiche.

4. Dans le champ **Folder/File**, saisissez le chemin d'accès au fichier source que vous souhaitez que le Job lise. Vous pouvez également parcourir votre système de fichiers afin de définir ce chemin.

   Si le fichier ne se trouve pas dans le système HDFS utilisé, vous devez le placer dans le système HDFS. Pour cela, vous pouvez par exemple utiliser le **tFileInputDelimited** et le **tHDFSOutput** dans un Job **Standard**.

Vérifier les composants de transformation

1. Double-cliquez sur le **tSortRow** afin d'ouvrir sa vue **Component**.

Ce composant garde la configuration utilisée dans le Job d'origine. Il trie les données d'entrée par ordre alphabétique à partir des colonnes *FirstName* et *LastName*.

2. Double-cliquez sur le **tUniqRow** pour ouvrir sa vue **Component**.

Le composant garde également la configuration utilisée dans le Job d'origine. Il sépare les données d'entrées en un flux **Uniques** et un flux **Duplicates** (Doublons), puis envoie les données uniques dans le **tHDFSOutput** et les doublons dans le **tJDBCOutput**.

Configurer le tHDFSOutput

1. Double-cliquez sur le **tHDFSOutput** pour ouvrir sa vue **Component**.

2. Comme expliqué précédemment pour la vérification du schéma du **tHDFSInput**, procédez de la même manière afin de vérifier le schéma du **tHDFSOutput**. S'il ne correspond pas au schéma du composant précédent, le **tUniqRow**, cliquez sur **Sync column** afin de récupérer le schéma du **tUniqRow**.

3. Dans le champ **Folder**, saisissez le chemin d'accès au dossier dans lequel vous souhaitez écrire les données uniques. Vous pouvez également parcourir votre système de fichiers afin de définir ce chemin.

4. Dans la liste **Action**, sélectionnez l'opération que vous souhaitez exécuter sur le dossier en question. Si le dossier existe déjà, sélectionnez **Overwrite**. Sinon, sélectionnez **Create**.

Configurer le tJDBCOutput

1. Double-cliquez sur le **tJDBCOutput** afin d'ouvrir sa vue **Component**.

2. Dans le champ **JDBC URL**, saisissez l'URL de la base de données dans laquelle vous souhaitez écrire les doublons. Dans cet exemple, l'URL est *jdbc:mysql://10.42.10.13:3306/Talend*, une base de données MySQL appelée *Talend*.

3. Dans le tableau **Drive JAR**, ajoutez une nouvelle ligne au tableau en cliquant sur le bouton .

4. Cliquez sur cette nouvelle ligne, puis cliquez sur le bouton afin d'ouvrir la boîte de dialogue **[Select Module]**. Cette boîte de dialogue vous permet d'importer le fichier Jar requis par la base de données MySQL.

5. Dans le champ **Class name**, saisissez la classe de fichier à appeler. Dans cet exemple, saisissez *org.gjt.mm.mysql.Driver*.

6. Dans les champs **User name** et **Password**, saisissez les informations d'authentification relatives à cette base de données.

7. Dans le champ **Table name**, saisissez le nom de la table dans laquelle vous souhaitez écrire les données, par exemple *Namelist*. Cette table doit exister.

Exécuter le Job

Appuyez sur la touche **F6** pour exécuter le Job.

Une fois le Job exécuté, vous pouvez visualiser les résultats d'exécution dans la console Web de HDFS et dans la base de données MySQL.

Dans HDFS, les entrées uniques sont écrites dans des fichiers séparés.

Id	FirstName	LastName	Age	City
4	Harry	Ford	48	Olympia
10	Grover	McKinley	33	Lansing

Dans MySQL, deux données dupliquées sont ajoutées.

Si vous souhaitez plus d'informations concernant le Job et son exécution, utilisez la console Web fournie par la distribution Hadoop que vous utilisez.

tUniservBTGeneric

Propriétés du tUniservBTGeneric

Famille de composant	Data Quality	
Fonction	Le composant **tUniservBTGeneric** permet l'exécution d'un processus (Job) créé avec le produit Uniserv *DQ Batch Suite*.	
Objectif	Le **tUniservBTGeneric** envoie les données à *DQ Batch Suite* et lance le processus spécifié dans *DQ Batch Suite*. Quand l'exécution du processus est terminée, les résultats sont renvoyés à *Data Quality Service Hub Studio* pour d'autres opérations.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Cliquez sur **Retrieve Schema** pour créer un schéma qui pour les composants qui correspondra aux champs d'entrée et de sortie du processus de *DQ Batch Suite*.
	Host name	Nom du serveur maître sur lequel est lancé *DQ Batch Suite*, entre guillemets doubles.
	Port	Numéro du port sur lequel le serveur de *DQ Batch Suite* est lancé, entre guillemets doubles.
	Client Server	Nom du serveur client de *DQ Batch Suite*, entre guillemets doubles.
	User name	Nom d'utilisateur enregistré pour le serveur de *DQ Batch Suite*. L'utilisateur doit avoir les droits pour exécuter le Job dans *DQ Batch Suite*.
	Password	Mot de passe de l'utilisateur. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Job directory	Répertoire de *DQ Batch Suite* dans lequel est sauvegardé le processus.
	Job name	Nom du processus *DQ Batch Suite* à exécuter.

	Job file path	Chemin d'accès à l'emplacement où le processus exécuté sera sauvegardé. Le chemin vers le fichier doit être absolu.
Advanced settings	*Temporary directory*	Répertoire dans lequel les fichiers temporaires créés lors de l'exécution du processus sont sauvegardés.
	Input Parameters	Ces paramètres doivent correspondre aux paramètres de la fonction **Input** de l'onglet **Format** du processus. **File location** : Définit si le fichier d'entrée est sauvegardé dans le pool ou dans le répertoire du processus local. **Directory** : Si **File location = Pool**, le répertoire est lié au pool. Si **File location = Job**, *"input"* doit être ici spécifié. **File name** : Nom du fichier délimiteur généré par le composant **tUniservBTGeneric** et transféré à *DQ Batch Suite*. Le nom du fichier doit correspondre au nom de fichier défini dans la fonction **Input** du processus. **No. of header rec.** : 0 = pas d'en-tête enregistré, 1 = en-tête dans le fichier d'entrée. **Field separator** : Séparateur de champs défini dans la fonction **Input** du processus *DQ Batch Suite*.
	Output Parameters	Ces paramètres doivent correspondre aux paramètres de la fonction **Output** de l'onglet **Format** du processus. **File location** : Définit si le fichier d'entrée est sauvegardé dans le pool ou dans le répertoire du processus local. **Directory** : Si **File location = Pool**, le répertoire est basé dans le pool. Si **File location = Job**, *"output"* doit être ici spécifié. **File name** : Nom du fichier de sortie au format délimiteur généré par le composant **tUniservBTGeneric** et transféré à *DQ Batch Suite*. Le nom du fichier doit correspondre au nom de fichier défini dans la fonction **Output** du processus. **No. of header rec.** : 0 = pas d'en-tête enregistré, 1 = en-tête dans le fichier de sortie. **Field separator** : Séparateur de champs défini dans la fonction **Output** du processus.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le composant **tUniservBTGeneric** envoie des données à *DQ Batch Suite* et lance le Job spécifié. Quand l'exécution du Job est terminée, les données de sortie sont renvoyées à *Data Quality Service Hub Studio* pour d'autres opérations.	
Limitation	Pour utiliser le composant **tUniservBTGeneric**, le logiciel Uniserv *DQ Batch Suite* doit être installé. Veuillez noter que : • le processus doit être configuré et exécutable dans *DQ Batch Suite*, • l'utilisateur doit posséder les droits pour exécuter le processus, • le processus *DQ Batch Suite* ne doit être composé que d'une seule ligne,	

	• les fichiers définis dans les fonctions **Input** et **Output** doivent posséder l'enregistrement du format délimiteur, • les données d'entrée et de sortie doivent être fournies selon le jeu de caractères UTF-8.

Scénario : Exécution d'un processus dans *DQ Batch Suite*

Ce scénario décrit un processus dont les résultats de l'exécution sont traités dans *Data Quality Service Hub Studio*. La source d'entrée du processus est fournie par *Data Quality Service Hub Studio*.

Le processus a été totalement défini dans *DQ Batch Suite* et sauvegardé sous le nom *"BTGeneric_Sample"*. Dans la fonction **Input**, le fichier *"btinput.csv"* a été spécifié comme fichier d'entrée sauvegardé dans le répertoire du Job et tous ses champs ont été renseignés. Le fichier n'est pas encore créé, car il sera fourni par *Data Quality Service Hub Studio*. Le processus n'est donc pas encore exécutable.

Dans *Data Quality Service Hub Studio*, la source d'entrée (dans cet exemple une table d'une base de données Oracle) de ce scénario est déjà enregistrée dans le **Repository**, de manière à ce que tous les schémas de métadonnées soient disponibles.

1. Dans la vue **Repository**, développez le nœud **Metadata** et le répertoire dans lequel vous avez sauvegardé la source. Glissez-la ensuite dans l'espace de modélisation graphique.

   La boîte de dialogue ci-dessous apparaît.

2. Sélectionnez **tOracleInput** et cliquez sur **OK** pour fermer la boîte de dialogue.

   Le composant s'affiche dans l'espace de modélisation graphique. La table utilisée dans ce scénario s'appelle *LOCATIONS*.

3. Glissez les composants suivants de la **Palette** dans l'espace de modélisation graphique : deux **tMap**, un **tOracleOutput** et un **tUniservBTGeneric**.

4. Connectez le **tMap** au **tUniservBTGeneric** à l'aide d'un lien **Row** > **Main**.

Acceptez le schéma du **tUniservBTGeneric** en cliquant sur **Yes** dans la fenêtre de propagation.

5. Connectez les autres composants à l'aide de liens **Row** > **Main**.

6. Double-cliquez sur le **tUniservBTGeneric** pour afficher sa vue **Basic settings**.

7. Saisissez les informations de connexion. Assurez-vous que le chemin d'accès que vous saisissez dans le champ **Job File Path** est absolu.

8. Cliquez sur **Retrieve Schema** pour créer automatiquement un schéma pour le **tUniservBTGeneric** à partir des définitions d'entrée et de sortie du processus et de renseigner automatiquement les champs de la vue **Advanced settings**.

9. Vérifiez que les informations de la vue **Advanced settings** soient correctes. Les définitions d'entrée et de sortie doivent être exactement identiques à celles du processus *DQ Batch Suite*. Si nécessaire, adaptez le chemin d'accès aux fichiers temporaires.

10. Double-cliquez sur le **tMap_1** pour ouvrir la fenêtre de mapping du schéma. À gauche se trouve la structure de la source d'entrée et à droite se trouve le schéma du **tUniservBTGeneric** (et donc l'entrée du processus). En bas se trouve le **Schema editor**, où vous pouvez trouver et éditer les attributs de chaque colonne.

11. Mappez les colonnes de la source d'entrée à leurs colonnes du **tUnivservBTGeneric** respectives. Pour ce faire, sélectionnez une colonne de la source d'entrée et glissez-la dans la colonne du même nom du côté droit.

Cliquez sur **OK** pour fermer la boîte de dialogue.

12. Définissez comment traiter les résultats d'exécution du processus et quels composants vont être utilisés.

13. Avant d'exécuter le processus, assurez-vous que les chemins d'accès sont corrects, que le serveur de *DQ Batch Suite* fonctionne et que vous pouvez accéder au processus.

tUniservRTConvertName

Propriétés du tUniservRTConvertName

Famille de composant	Data Quality	
Fonction	**tUniservRTConvertName** analyse un nom complet par rapport au contexte. S'il s'agit de personnes, il divise le nom en segments (nom, prénom, civilité, préfixes, suffixes etc.) et créer une clé d'adresse. Le composant reconnaît les adresses d'entreprise ou d'institution et est capable de fournir la fiche de l'organisme séparément. Il divise également les lignes contenant des informations sur plusieurs personnes. De plus, il reconnaît certains modèles n'appartenant pas aux informations de nom, dans la ligne de nom (numéro client, notes, etc.) et les supprime ou les déplace dans certains champs mémo spécifiques.	
Objectif	Le **tUniservRTConvertName** fournit une base pour une structure et une population uniformes de noms de personnes et d'entreprises dans la base de données, ainsi que des titres personnalisés.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Host name	Nom de l'hôte du serveur, entre guillemets doubles.
	Port	Numéro du port du serveur, entre guillemets doubles.
	Service	Le nom/type du service est `"cname_d"` par défaut. Saisissez un nouveau nom si nécessaire (si le suffixe du service change, par exemple), entre guillemets doubles. Les services disponibles sont : Allemagne `"cname_d"` Italie `"cname_i"` Autriche `"cname_a"` Pays-Bas `"cname_nl"` Suisse `"cname_ch"` Belgique `"cname_b"` France `"cname_f"` Espagne `"cname_e"`

	Use rejects	Cochez cette case pour écrire séparément en sortie les ensembles de données séparément d'une certaine classe de résultats lors de l'analyse des noms. Saisissez la classe respective des résultats dans le champ **if result class is greater or equal to**. Si cette case n'est pas cochée, les ensembles sont tout de même écrits en sortie via la connexion **Main**, même si l'analyse a échoué. Si cette case est cochée mais que la connexion **Rejects** n'a pas été établie, les ensembles sont simplement triés quand l'analyse échoue.
Advanced settings	*Analysis Configuration*	Pour plus d'informations, veuillez consulter le manuel utilisateur d'Uniserv *convert-name*.
	Output Configuration	Pour plus d'nformations, veuillez consulter le manuel utilisateur d'Uniserv *convert-name*.
	Configuration of not recognized input	Pour plus d'informations, veuillez consulter le manuel utilisateur d'Uniserv *convert-name*.
	Configuration of free fields	Pour plus d'informations, veuillez consulter le manuel utilisateur d'Uniserv *convert-name*.
	Cache Configuration	Pour plus d'informations, veuillez consulter le manuel utilisateur d'Uniserv *convert-name*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le composant **tUniservRTConvertName** fournit une base pour une structure et une population uniformes de noms de personnes et d'entreprises dans la base de données, ainsi que des titres personnalisés.	
Limitation	Pour utiliser le **tUniservRTConvertName**, le logiciel Uniserv doit être installé.	

Scénario : Analyse d'une ligne de nom et attribution d'un titre

Ce scénario décrit un Job vous permettant d'analyser le nom de personnes dans un fichier et de leur attribuer un titre.

Le fichier d'entrée de ce scénario est déjà sauvegardé dans le **Repository**, toutes les métadonnées du schéma sont donc disponibles.

Veuillez noter que les données d'une source d'entrée doivent toutes être liées au même pays.

1. Dans la vue **Repository**, déroulez le nœud **Metadata** et le répertoire dans lequel est sauvegardé le fichier. Glissez ce fichier dans l'espace de modélisation graphique.

   La boîte de dialogue suivante s'ouvre :

2. Sélectionnez le composant **tFileInputDelimited** et cliquez sur **OK** pour fermer la boîte de dialogue.

   Le composant s'affiche dans l'espace de modélisation graphique. Le fichier utilisé dans ce scénario est appelé *SampleAddresses.*.

3. Glissez les composants suivants de la **Palette** à l'espace de modélisation graphique : deux composants **tMap**, un **tUniservRTConvertName,** et un**tFileOutputDelimited**.

4. Reliez le **tMap** au **tUniservRTConvertName** à l'aide d'un lien **Row** > **Main**.

   Acceptez le schéma du **tUniservRTConvertName** en cliquant sur **Yes** dans la fenêtre de propagation.

5. Connectez les autres composants à l'aide de liens **Row** > **Main**.

6. Double-cliquez sur le **tMap_1** pour ouvrir la fenêtre de mapping. À gauche se trouve la structure de la source d'entrée et à droite se trouve le schéma du **tUniservRTConvertName** (et donc l'entrée du Job). En bas se trouve le **Schema editor**, où vous pouvez trouver et éditer les attributs de chaque colonne.

7. Assignez les colonnes de la source d'entrée à leurs colonnes respectives du **tUnivservRTConvertName**. Pour ce faire, sélectionnez une colonne de la source d'entrée et glissez-la dans la colonne du même nom du côté droit. Si les champs du fichier d'entrée doivent être passés au fichier de sortie, comme les adresses ou les IDs, vous devrez définir des champs supplémentaires.

8. Cliquez sur **OK** pour fermer la boîte de dialogue.

9. Double-cliquez sur le **tUniservRTConvertName** pour ouvrir sa vue **Basic settings**.

10. Renseignez les informations concernant le serveur et spécifiez le pays lié au service.

11. Double-cliquez sur le **tMap_3** pour ouvrir la fenêtre de mapping. À gauche se trouve le schéma du **tUniservRTConvertName** (et donc la sortie du Job) et à droite le schéma du fichier de sortie.

12. Cliquez sur **OK** pour fermer la fenêtre.
13. Double-cliquez sur le **tFileOutputDelimited** et saisissez les détails du fichier de sortie.

tUniservRTMailBulk

Propriétés du tUniservRTMailBulk

Famille de composant	Data Quality	
Fonction	Le **tUniservRTMailBulk** créé un pool d'index pour *mailRetrieval* avec des données d'entrée prédéfinies.	
Objectif	Le **tUniservRTMailBulk** prépare le pool d'index pour la recherche de doublons.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in) soit distant dans le **Repository**. Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur **Edit schema** afin de visualiser le schéma.
	Host name	Nom de l'hôte du serveur, entre guillemets doubles.
	Port	Numéro du port d'écoute du serveur, entre guillemets doubles.
	Service	Le nom du service est `"mail"` par défaut. Saisissez une nouvelle valeur si nécessaire (si le suffixe du service change, par exemple), entre guillemets doubles.
Advanced settings	*Uniserv Parameters*	Pour plus d'informations, veuillez consulter le manuel utilisateur Uniserv *mailRetrieval*.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job et de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Usage	Le **tUniservRTMailBulk** prépare le pool d'index pour une recherche de doublons.	
Limitation	Pour utiliser le composant **tUniservRTMailBulk**, le logiciel Uniserv doit être installé. Vous aurez besoin d'un composant d'entrée et d'un composant de mapping pour lire l'adresse provenant d'une base de données ou d'un fichier. Le composant ne possède pas de connexion de sortie.	

Scénario : Créer un pool d'index

Ce scénario décrit un Job chargeant une liste d'adresses d'une base de données SQL vers un pool d'index.

La base de données de ce scénario est déjà sauvegardée dans le **Repository**, toutes les métadonnées de schéma sont donc déjà disponibles.

1. Dans la vue **Repository**, développez le nœud **Metadata** et le répertoire dans lequel est sauvegardé le fichier. Glissez ce fichier dans l'espace de modélisation graphique.

   La boîte de dialogue suivante s'ouvre :

2. Sélectionnez le composant **tMysqlInput** et cliquez sur **OK** pour fermer la boîte de dialogue.

   Le composant apparaît dans l'espace de modélisation graphique.

3. Glissez les composants suivants de la **Palette** dans l'espace de modélisation graphique: **tMap** et **tUniservRTMailBulk**.

4. Connectez le **tMap** au **tUniservRTMailBulk** à l'aide d'un lien **Row** > **Main**.

   Acceptez le schéma du **tUniservRTMailBulk** en cliquant sur **Yes** dans la fenêtre de propagation.

5. Connectez les autres composants à l'aide de liens **Row** > **Main**.

6. Double-cliquez sur le **tMap_1** pour ouvrir la fenêtre de mapping. À gauche se trouve le schéma du fichier de la base de données et à droite se trouve le schéma du **tUniservRTMailBulk** (et donc l'entrée du Job). En bas se trouve le **Schema editor**, où vous pouvez trouver et éditer les attributs de chaque colonne.

7. Assignez les colonnes de la source d'entrée à leurs colonnes du **tUniservRTMailBulk** respectives. Pour ce faire, sélectionnez une colonne de la source d'entrée et glissez-la dans la colonne du même nom du côté droit. La signification de chaque argument est décrite dans le manuel utilisateur Uniserv *mailRetrieval*.

8. Cliquez sur **OK** pour fermer la fenêtre de dialogue.

9. Double-cliquez sur le **tUniservRTMailBulk** pour ouvrir sa vue **Basic settings**.

10. Renseignez les informations relatives au serveur et spécifiez le service.

11. Dans la vue **Advanced settings**, adaptez les paramètres serveur.

tUniservRTMailBulk_1
Basic settings
Advanced settings
Dynamic settings
View
Documentation
Uniserv parameters
par_create_pool
UNI_FALSE
par_date_format
"dd.mm.yyyy"
tStatCatcher Statistics

tUniservRTMailOutput

Propriétés du tUniservRTMailOutput

Famille de composant	Data Quality	
Fonction	Le **tUniservRTMailOutput** met à jour le pool d'index utilisé pour les recherches de doublons.	
Objectif	La **tUniservRTMailOutput** garde le pool d'index synchronisé.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Host name	Nom de l'hôte du serveur, entre guillemets doubles.
	Port	Numéro du port d'écoute du serveur, entre guillemets doubles.
	Service	Le nom du service est `"mail"` par défaut. Saisissez une nouveau nom si nécessaire (si le suffixe du service change, par exemple), entre guillemets doubles.
	Action on data	Opérations réalisables dans le pool d'index : **Insert** : insère un nouvel enregistrement dans le pool d'index. Cette requête échoue si l'enregistrement possédant la même référence existe déjà dans le pool. **Update** : met à jour un enregistrement existant dans le pool. Cette requête échoue si aucun enregistrement ne possède le nom de la référence donnée. **Insert or update** : insère un nouvel enregistrement dans le pool d'index. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas dans le pool d'index, un nouvel enregistrement est inséré.. **Delete** : supprime l'enregistrement correspondant à la référence donnée dans le pool d'index.
Advanced settings	*Uniserv Parameters*	Pour plus d'informations, consultez le manuel utilisateur d'Uniserv *mailRetrieval*.

	tStatCatcher Statistics	Cochez cette case pour collecter les données d'identification au niveau du Job et du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le **tUniservRTMailOutput** met à jour les pool d'index et passe l'ensemble d'entrée. La sortie contient le statut de l'opération. Si l'opération échoue, un message d'erreur message s'affiche.	
Limitation	Pour utiliser le **tUniservRTMailOutput**, le logiciel Uniserv *mailRetrieval* doit être installé. Avant d'utiliser le **tUniservRTMailOutput** pour la première fois, un pool d'index doit être créé. Créez-en un à l'aide du composant **tUniservRTMailBulk**.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Ajouter des contacts au pool d'index de mailRetrieval*.

tUniservRTMailSearch

Propriétés du tUniservRTMailSearch

Famille de composant	Data Quality	
Fonction	Le composant **tUniservRTMailSearch** recherche des données similaires dans l'enregistrement d'entrée donné.	
Objectif	Le **tUniservRTMailSearch** cherche les valeurs dupliquées et ajoute de nouvelles données à chaque enregistrement.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Host name	Nom de l'hôte du serveur, entre guillemets doubles.
	Port	Numéro du port d'écoute du serveur, entre guillemets doubles.
	Service	Le nom du service est `"mail"` par défaut. Saisissez une nouveau nom si nécessaire (si le suffixe du service change, par exemple), entre guillemets doubles.
	Maximum of displayed duplicates (0 = All)	Saisissez le nombre maximum de doublons à afficher dans la vue **Run**. La valeur par défaut est 0, ce qui signifie que tous les doublons sont affichés. Mille doublons peuvent être affichés au maximum.
	Use rejects	Cochez cette case pour paramétrer les options définissant les enregistrements de doublons ajouter au flux de rejet. Définissez : **Element** : Compte des doublons. **Operator** : <, <=, =, >=, >. **Value** : Saisissez la valeur manuellement..
Advanced settings	*Uniserv Parameters*	Pour plus d'informations, veuillez consulter le manuel utilisateur Uniserv *mailRetrieval*.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job et de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Le **tUniservRTMailSearch** requiert un composant d'entrée et un ou plusieurs composant(s) de sortie.
Limitation	Pour utiliser le composant **tUniservRTMailSearch**, le logiciel Uniserv *mailRetrieval* doit être installé. Avant d'utiliser le **tUniservRTMailSearch**, un pool d'index doit être créé. Vous pouvez créer un pool d'index grâce au composant **tUniservRTMailBulk**.

Scénario : Ajouter des contacts au pool d'index de mailRetrieval

Ce scénario décrit un Job qui ajoute des contacts à un pool d'index de *mailRetrieval*. Vous devez au préalable vous assurer que ces contacts existent déjà.

Le fichier d'entrée de ce scénario est sauvegardé dans le **Repository**, ce qui rend disponible le schéma des métadonnées.

Les données de la source d'entrée doivent provenir du même pays.

Déposer et relier les composants

1. Dans la vue **Repository**, développez le nœud **Metadata** et le répertoire dans lequel le fichier est sauvegardé. Glissez ce fichier dans l'espace de modélisation graphique.

   La boîte de dialogue suivante apparaît :

2. Sélectionnez **tFileInputDelimited** et cliquez sur **OK** pour fermer la boîte de dialogue.

Le composant s'affiche dans l'espace de modélisation graphique.

3. Glissez les composants suivants de la **Palette** dans l'espace de modélisation graphique : deux composants **tMap**, un **tUniservRTMailSearch** et un **tUniservRTMailOutput**.

4. Connectez le **tMap** au **tUniservRTMailSearch** à l'aide d'un lien **Row** > **Main**.

   Acceptez le schéma du **tUniservRTMailSearch** en cliquant sur **Yes** dans la fenêtre de propagation.

5. Connectez les autres composants à l'aide de liens **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le composant **tMap_1** pour ouvrir la fenêtre de mapping de son schéma. Sur la gauche se trouve la structure du fichier d'entrée et sur la droite le schéma du **tUniservRTMailSearch**. En bas se trouve le **Schema Editor**, où vous pouvez trouver et éditer les attributs de chaque colonne.

2. Mappez les colonnes du fichier d'entrée à leurs colonnes respectives dans le **tUniservRTMailSearch**. Pour ce faire, sélectionnez une colonne de la source d'entrée et glissez-la sur la colonne appropriée du côté droit.

3. Quand une liste d'entrée contient un ID de référence, il est conseillé de l'accepter. Pour ce faire, créez une nouvelle colonne appelée *IN_DBREF* depuis le **Schema Editor** et connectez-la à votre ID de référence.

   Cliquez sur **OK** pour fermer la fenêtre.

4. Double-cliquez sur le composant **tUniservRTMailSearch** pour ouvrir sa vue **Basic settings**.

5. Dans la case **Maximum of displayed "duplicates"**, saisissez *0* pour afficher tous les doublons.

   Sélectionnez **Define rejects** pour ouvrir la fenêtre de définition des rejets.

6. Cliquez sur le bouton **[+]** pour insérer une ligne dans la fenêtre. Sélectionnez *Duplicate count* dans la colonne **Elements**, > dans la colonne **Operator** et *0* dans la colonne **Value**. De cette façon, tous les contacts existants sont écartés et seuls les nouveaux contacts sont ajoutés au pool d'index.

7. Cliquez sur l'onglet **Advanced settings** et vérifiez les paramètres. Des paramètres par défaut ont été définis. Référez-vous au manuel de *mailRetrieval* pour plus de précisions.

8. Double-cliquez sur le second composant **tMap** pour ouvrir la fenêtre de mapping de son schéma. Sur la gauche se trouve le schéma du **tUniservRTMailSearch** et sur la droite celui du **tUniservRTMailOutput**.

9. Cliquez sur **Auto map!** pour mapper les champs automatiquement.

10. Le seul champ devant être mappé manuellement est l'ID de référence. Pour ce faire, glissez *OUT-DBREF* vers le champ *IN_DBREF* du côté droit.

Cliquez sur **OK** pour fermer la boîte de dialogue.

11. Double-cliquez sur le composant **tUniservRTMailOutput** pour afficher sa vue **Basic settings**.

12. Dans la liste **Action on Data**, sélectionnez *Insert or update* pour que tous les nouveaux contacts soient ajoutés au pool d'index.

tUniservRTPost

Propriétés du tUniservRTPost

Famille de composant	Data Quality	
Fonction	Le composant **tUniservRTPost** fournit un service de validation postale et de correction d'adresses, critique pour améliorer la qualité des adresses. Grâce à ce composant, vous serez plus efficace dans le marketing individualisé, réduisant les coûts et augmentant l'efficacité de la gestion des adresses dans toutes les applications.	
Objectif	Le composant **tUniservRTPost** est conçu pour améliorer la qualité des adresses, ce qui est extrêmement important pour la GRC et le e-commerce : cela est directement lié aux coûts du courrier et de la publicité.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host name	Nom de l'hôte du serveur, entre guillemets doubles.
	Port	Numéro du port d'écoute du serveur, entre guillemets doubles.
	Service	Le nom du service est `"post"` par défaut. Saisissez un nouveau nom si nécessaire (suffixe du service par exemple), entre guillemets doubles.
	Use rejects	Cochez cette case pour collecter les adresses incorrectes via la connexion **rejects**. Il s'agit habituellement des adresses de classe 5. Les valeurs valides pour les classes de résultat vont de 1 à 5. La valeur doit être entre guillemets doubles. Si cette case n'est pas cochée, les adresses incorrectes sont écrites en sortie via la connexion **Main**. Si cette case est cochée mais que la connexion **rejects** n'est pas créée, les adresses incorrectes sont simplement rejetées.
	Use File for ambiguous results	Cochez cette case pour définir un fichier qui servira à écrire la liste des résultats ambigus.

		Lorsqu'une adresse ne peut être corrigée de manière certaine, une liste de résultats ambigus est créée. Cette liste peut par la suite être traitée via la connexion **AMBIGUITY**. Tous les résultats potentiellement candidats sont redirigés via cette connexion. Le schéma de cette connexion est pré-initialisé avec les arguments de la liste de sélection dissoute du service *'post'*.
Advanced settings	*Uniserv Parameters*	Cochez cette case pour définir les paramètres correspondants. Pour plus d'informations, veuillez vous référer au manuel Uniserv *International Postal Framework*.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job et de chaque composant.
	"Full address" selection list	Cochez la case **Display** pour afficher toutes les colonnes. Vous pouvez également cocher la case d'une colonne en particulier pour n'afficher qu'elle. Cette option contrôle le contenu du fichier des adresses incertaines. Seules les colonnes sélectionnées seront écrites dans le fichier.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le composant **tUniservRTPost** requiert une liste d'adresses en entrée. Les données concernant les adresses sont ensuite vérifiées. Si la liste ne contient que des résultats non ambigus, elle est écrite en sortie via la connexion **Main**. Si l'adresse est ambigüe, les candidats potentiels sont écrits en sortie via la connexion **Ambiguity**. Si une adresse n'a pas pu être trouvée, elle sera transférée via la connexion **Reject**.	
Limitation	Pour utiliser le composant **tUniservRTPost**, le logiciel Uniserv *International Postal Framework* et les serveurs dédiés doivent être installés.	

Scénario 1 : Vérifier et corriger le code postal, la ville et la rue

Ce scénario décrit un Job qui vérifie et corrige les adresses et les codes postaux d'un fichier.

Le fichier d'entrée de ce scénario est déjà sauvegardé dans le **Repository**, les schémas de métadonnées sont donc disponibles.

1. Dans la vue **Repository**, développez le nœud **Metadata** et le répertoire dans lequel le fichier est sauvegardé. Glissez ce fichier dans l'espace de modélisation graphique.

   La boîte de dialogue suivante apparaît :

2. Sélectionnez le composant **tFileInputDelimited** et cliquez sur **OK** pour fermer la boîte de dialogue.

   Le composant s'affiche dans l'espace de modélisation graphique. Le fichier utilisé dans ce scénario se nomme *SampleAddresses*. Il contient des adresses comprenant un code pays. La rue et le numéro sont sauvegardés ensemble dans le champ *Street* et le code postal et la ville sont sauvegardés dans des champs séparés.

3. Glissez les composants suivants de la **Palette** dans l'espace de modélisation graphique : deux composants **tMap**, un composant **tUniservRTPost** et un **tFileOutputDelimited**.

4. Connectez le **tMap** au **tUniservRTPost** à l'aide d'un lien **Row** > **Main**.

   Acceptez le schéma du **tUniservRTPost** en cliquant sur **Yes** dans la fenêtre proposant la propagation.

5. Connectez les autres composants à l'aide de liens **Row** > **Main**.

6. Double-cliquez sur le **tMap_1** pour ouvrir la fenêtre de mapping du schéma. Sur la gauche se trouve la structure du fichier d'entrée et sur la droite le schéma du **tUniservRTPost**. En bas se trouve le **Schema Editor**, où vous pouvez trouver et éditer les attributs de chaque colonne.

7. Mappez les colonnes du fichier d'entrée à leurs colonnes respectives dans le **tUniservRTPost**. Pour ce faire, sélectionnez une colonne de la source et glissez-la sur la colonne appropriée du côté droit. Si des champs du fichier d'entrée doivent être passés au fichier de sortie, comme les noms ou les IDs, de nouveaux champs doivent être définis.

   Au moment de mapper les champs, notez que la rue et le numéro peuvent être enregistrés soit ensemble dans la colonne *Street*, soit dans des champs séparés. Si votre liste d'adresses ne comporte pas de code pays mais que les adresses proviennent d'un même pays, vous devrez saisir manuellement le code ISO du pays correspondant entre guillemets doubles dans la colonne *IN_COUNTRY*. Dans le cas d'une liste d'adresses internationales sans code pays, laissez la colonne *IN_COUNTRY* vide. Pour plus d'informations, veuillez vous référer au guide utilisateur Uniserv *International Postal Framework*.

8. Cliquez sur **OK** pour fermer la fenêtre.

9. Double-cliquez sur le composant **tUniservRTPost** et ouvrez sa vue **Advanced settings**.

10. Changez les paramètres et les longueurs de champs si nécessaire et sélectionnez les champs de sortie.

Assurez-vous d'avoir défini une longueur suffisante de champ. Pour plus de précisions, veuillez vous référer au Guide utilisateur Uniserv *International Postal Framework*.

11. Double-cliquez sur le second composant **tMap** pour ouvrir la fenêtre de mapping du schéma. Sur la gauche se trouve le schéma du **tUniservRTPost** et sur la droite le schéma du fichier de sortie.

12. Cliquez sur **OK** pour fermer la boîte de dialogue.

13. Double-cliquez sur le composant **tFileOutputDelimited** pour saisir les détails du fichier de sortie.

Scénario 2 : Vérifier et corriger le code postal, la ville et la rue, tout en rejetant les données incorrectes

Ce scénario est étroitement lié au précédent à ceci près que les adresses qui ne peuvent être assignées sont écrites dans un champ à part pour une vérification manuelle. De plus, pour écrire des adresses ambiguës dans un fichier séparé, la procédure est la même que celle décrite ici.

1. Créez un Job comme décrit dans le scénario précédent.

2. Glissez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tMap** et un **tFileOutputDelimited**.

3. Double-cliquez sur le composant **tUniservRTPost** pour ouvrir sa vue **Basic settings**.

4. Cochez la case **Use rejects** et saisissez *"5"* dans le champ **if result class greater or equals to**. Il s'agit de la classe de résultat de la vérification des codes postaux dans les adresses contenant des données incorrectes ou insuffisantes.

5. Connectez le composant **tUniservRTPost** au **tMap_5** à l'aide d'un lien **Row** > **Rejects**.

6. Connectez le **tMap** au **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main**.

7. Définissez les champs du fichier de sortie dans la fenêtre de mapping du schéma.

Databases - Composants traditionnels

Le chapitre *Databases - Composants traditionnels* décrit les connecteurs de bases de données traditionnelles ainsi que les connecteurs aux bases de données les plus populaires. Ces connecteurs couvrent divers besoins : ouverture de connexion, lecture et écriture de tables, commit de transactions, ainsi que des rollbacks pour le traitement des erreurs. Plus de 40 SGBDR sont supportés. Ces composants sont présents dans la famille **Databases** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

D'autres types de connecteurs de bases de données, tels que les connecteurs d'appliance, de datawarehouse et de gestion de bases de données, sont documentés dans le *Databases - Composants d'appliance et de datawarehouse* et le *Databases - Autres composants*.

tAccessBulkExec

Propriétés du tAccessBulkExec

Les composants **tAccessOutputBulk** et **tAccessBulkExec** sont généralement utilisés ensemble pour écrire des données dans un fichier délimité puis effectuer différentes actions sur le fichier dans une base de données Access, processus en deux étapes. Ces deux étapes sont regroupées dans le composant **tAccessOutputBulkExec**, détaillé dans une section séparée. L'intérêt d'utiliser un processus en deux étapes réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/ Access	
Fonction	Le composant **tAccessBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tAccessBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Access	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version d'Access que vous utilisez.
	Database	Nom de la base de données.
	Username and *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.

		Clear table : supprimer le contenu de la table.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit exister pour que l'opération Insert soit autorisée.
	Local filename	Nom du fichier à traiter. Voir également le *Guide utilisateur de Studio Talend*.
	Action on data	Vous pouvez effectuer, sur les données de la table définie, l'opération que vous souhaitez : **Insert** : ajoute de nouvelles entrées à la table.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur de Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur de Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données existante. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tAccessOutputBulk**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de donnés Access.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Si vous utilisez la version 64 bits du Studio, vous devez utiliser la version 32 bits du pilote ODBC et de l'environnement Java.	

Scénarios associés

Pour plus d'informations relatives au fonctionnement du composant **tAccess BulkExec**, consultez les scénarios suivants :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du composant **tMysqlOutputBulkExec**.

tAccessClose

Propriétés du tAccessClose

Famille de composant	Databases/Access	
Fonction	Le composant **tAccessClose** ferme une connexion active à une base de données.	
Objectif	Le **tAccessClose** est utilisé pour arrêter une connexion à Access afin de libérer des ressources occupées.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tAccessConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Access, notamment avec le **tAccessConnection** et le **tAccessCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Si vous utilisez la version 64 bits du Studio, vous devez utiliser la version 32 bits du pilote ODBC et de l'environnement Java.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tAccessCommit

Propriétés du tAccessCommit

Famille de composant	Databases/Access	
Fonction	Le composant **tAccessCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tAccessConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tAccessCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Access et notamment avec les composants **tAccesConnection** et **tAccessRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Si vous utilisez la version 64 bits du Studio, vous devez utiliser la version 32 bits du pilote ODBC et de l'environnement Java.	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tAccessCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*

tAccessConnection

Propriétés du tAccessConnection

Famille de composant	Databases/Access	
Fonction	Le composant **tAccessConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez la version d'Access que vous utilisez (Access 2003 ou versions supérieures).
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec des composants Access et notamment avec les composants **tAccessCommit** et **tAccessRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Si vous utilisez la version 64 bits du Studio, vous devez utiliser la version 32 bits du pilote ODBC et de l'environnement Java.	

Scénario: Insérer des données dans des tables parent/enfant

Le Job suivant est dédié à des utilisateurs avancés de base de données qui souhaitent effectuer des insertions dans des tables multiples en utilisant une table parent *Table1* pour générer deux tables enfant : *Name* et *Birthday*.

- Créez une base de données connectée à Access 2007 que vous appellerez *Database1*.
- Une fois la base de données Access créée, créez une table appelée *Table1* avec les deux en-têtes de colonne suivants : *Name* et *Birthday*.

Dans la perspective **Integration** de votre *Studio Talend*, le Job est constitué de douze composants, parmi lesquels des composants **tAccessConnection**, **tAccessCommit**, **tAccessInput**, **tAccessOutput** et **tAccessClose**.

- A partir de la **Palette**, cliquez et déposez les composants suivants dans l'espace de modélisation graphique : un **tFileList**, un **tFileInputDelimited**, un **tMap**, deux **tAccessOutput**, deux **tAccessInput**, un **tAccessCommit**, un **tAccessClose** et deux **tLogRow**.
- Connectez le composant **tFileList** au composant d'entrée **tFileInputDelimited** à l'aide d'une connexion de type **Iterate**. Ainsi, le nom du fichier à traiter sera automatiquement renseigné à partir de la liste de fichiers du **tFileList** en utilisant une variable globale.
- Connectez le composant **tFileInputDelimited** au **tMap** et séparez le flux vers les deux composants de sortie **tAccessOutput**. Utilisez des connexions de type **Row** pour chacune de ces connexions représentant un flux de données principal.
- Définissez les propriétés du composant **tFileList**, notamment le répertoire dans lequel sont stockés les fichiers.
- Ajoutez le composant **tAccessConnection** et connectez-le au composant de départ de ce Job (dans cet exemple, le composant **tFileList**), à l'aide d'une connexion **OnComponentOk** afin de définir l'ordre d'exécution.

- Dans la vue **Component** du composant **tAccessConnection**, définissez manuellement les informations de connexion ou récupérez-les dans le **Repository** si vous les avez préalablement stockées dans le dossier **DB connection** du répertoire **Metadata**. Pour plus d'informations concernant les Métadonnées, consultez le *Guide utilisateur du Studio Talend*.

- Dans l'onglet **Basic settings** du composant **tFileInputDelimited**, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. Définissez le champ **File Name** avec la variable globale : *tFileList_1.CURRENT_FILEPATH*. Pour plus d'informations concernant l'utilisation des variables, consultez le *Guide utilisateur du Studio Talend*.

- Paramétrez les autres champs comme vous le souhaitez, définissez les séparateurs de lignes et de champs (**Row** et **Field Separator**) en fonction de la structure de votre document.

- Puis définissez manuellement le schéma du fichier à l'aide du bouton **Edit schema** ou sélectionnez le schéma dans le **Repository**. Assurez-vous que le type de données est correctement défini, conformément à la nature des données traitées.

- Dans la zone Output du **tMap**, ajoutez deux tables de sortie, une que vous nommerez *Name* pour la table *Name*, et une deuxième que vous nommerez *Birthday*, pour la table *Birthday*. Pour plus d'informations concernant le composant **tMap**, consultez le *Guide utilisateur du Studio Talend*.

- Cliquez sur la colonne *Name* de la zone **Input**, et glissez-la dans la table *Name*.

- Cliquez sur la colonne *Birthday* de la zone **Input**, et glissez-la dans la table *Birthday*.

- Puis sélectionnez une connexion de sortie de type **Row** pour acheminer correctement le flux vers les composants Database de sortie correspondants.

- Dans l'onglet **Basic settings** des deux composants **tAccessOutput**, cochez la case **Use an existing connection** pour récupérer les informations du composant **tAccessConnection**.

Ignorez le champ **Commit every** puisque cette commande sera annulée par le composant **tAccessCommit**.

- Nommez votre table dans le champ **Table** en vous assurant que vous saisissez la bonne table. Dans cet exemple, la table est soit *Name*, soit *Birthday*.
- Ne définissez aucune action dans le champ **Action on table** puisque les tables ont déjà été créées.
- Sélectionnez **Insert** dans le champ **Action on data** pour les deux composants de sortie (**tAccessOutput**).
- Cliquez sur le bouton **Sync columns** pour récupérer le schéma défini dans le **tMap**.
- Connectez ensuite le premier composant **tAccessOutput** au premier composant **tAccessInput** à l'aide d'une connexion **OnComponentOk**.
- Dans l'onglet **Basic settings** des deux composants **tAccessInput**, cochez la case **Use an existing connection** pour récupérer le flux de données distribué. Puis définissez manuellement le schéma du fichier à l'aide du bouton **Edit schema**.
- Nommez ensuite votre table dans le champ **Table Name**. Le nom du **tAccessInput_1** sera *Name*.
- Cliquez sur **Guess Query**.
- Connectez les deux composants **tAccessInput** au composant **tLogRow** à l'aide d'une connexion de type **Row** > **Main**. Dans l'onglet **Basic settings** des deux composants **tLogRow**, sélectionnez **Table** dans le champ **Mode**.
- Ajoutez le composant **tAccessConnection** sous le composant **tFileList** et connectez-les à l'aide d'une connexion **OnComponentOk** afin d'arrêter le Job avec les commits de transaction.
- A partir de l'onglet **Basic settings** du composant **tAccessCommit**, sélectionnez dans la liste déroulante **Component list** la connexion à utiliser, **tAccessConnection_1** dans ce scénario.
- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job Access at 18:02 28/12/2010.

[statistics] connecting to socket on port 3938
[statistics] connected
.----.
|tLogRow_1|
|=---=|
|Name|
|=---=|
|Amy  |
|Bob  |
'----'

.----------.
|tLogRow_2  |
|=---------=|
|Birthday   |
|=---------=|
|02-02-2011|
|03-03-2023|
'----------'

[statistics] disconnected
Job Access ended at 18:02 28/12/2010. [exit code=0]
```

La table parent *Table1* est utilisée à la fois pour générer la table *Name* et la table *Birthday*.

tAccessInput

Propriétés du tAccessInput

Famille de composant	Databases/Access	
Fonction	Le composant **tAccessInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tAccessInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez la section expliquant comment configurer une connexion à une base de données dans le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version d'Access que vous utilisez.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type et Query	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Trim all the String/ Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Si vous utilisez la version 64 bits du Studio, vous devez utiliser la version 32 bits du pilote ODBC et de l'environnement Java.	

Scénarios associés

Pour un scénario associé, consultez les scénarios **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Pour plus d'informations sur les contextes dynamiques dans le cadre d'une utilisation de base de données, consultez *tContextLoad*.

tAccessOutput

Propriétés du tAccessOutput

Famille de composant	Databases/Access	
Fonction	Le composant **tAccessOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tAccessOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version d'Access que vous utilisez.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.

	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets.

		Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tAccessOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Support null in "SQL WHERE" statement	Cochez cette case pour prendre en compte les valeurs Null d'une table de base de données. Assurez-vous que la case Nullable est bien cochée pour les colonnes du schéma correspondantes.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant.	

	La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Access. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Si vous utilisez la version 64 bits du Studio, vous devez utiliser la version 32 bits du pilote ODBC et de l'environnement Java.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du **tMySQLOutput**.

tAccessOutputBulk

Propriétés du tAccessOutputBulk

Les composants **tAccessOutputBulk** et **tAccessBulkExec** sont généralement utilisés ensemble pour écrire des données dans un fichier délimité puis effectuer différentes actions sur le fichier dans une base de données Access, processus en deux étapes. Ces deux étapes sont regroupées dans le composant **tAccessOutputBulkExec**, détaillé dans une section séparée. L'intérêt d'utiliser un processus en deux étapes réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/Access	
Fonction	Le composant **tAccessOutputBulk** écrit un fichier délimité.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Access.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Chemin d'accès et nom du fichier à traiter, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Include header*	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tAccessBulkExec**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données Access.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Si vous utilisez la version 64 bits du Studio, vous devez utiliser la version 32 bits du pilote ODBC et de l'environnement Java.	

Scénarios associés

Pour un scénario associé au **tAccessOutputBulk**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tAccessOutputBulkExec

Propriétés du tAccessOutputBulkExec

Les composants **tAccessOutputBulk** et **tAccessBulkExec** sont généralement utilisés ensemble afin d'écrire des données dans un fichier délimité, puis d'effectuer différentes actions sur le fichier dans une base de données Access, processus en deux étapes. Ces deux étapes sont regroupées dans le **tAccessOutputBulkExec**.

Famille de composant	Databases/Access	
Fonction	Le **tAccessOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tAccessOutputBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Access.	
Basic settings	*Property Type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version d'Access que vous utilisez.
	DB Name	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table.

	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'insert soit autorisée.
	File Name	Nom du fichier à traiter. Voir également le *Guide utilisateur du Studio Talend*.
	Action on data	Vous pouvez effectuer, sur les données de la table définie, l'opération que vous souhaitez : **Insert** : ajoute de nouvelles entrées à la table.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.
Limitation	Si vous utilisez la version 64 bits du Studio, vous devez utiliser la version 32 bits du pilote ODBC et de l'environnement Java.

Scénarios associés

Pour un scénario associé au **tAccessOutputBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tAccessRollback

Propriétés du tAccessRollback

Famille de composant	Databases/Access	
Fonction	Le **tAccessRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tAccessConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé en association avec des composants Access, notamment avec le **tAccessConnection** et le **tAccessCommit**.	
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Si vous utilisez la version 64 bits du Studio, vous devez utiliser la version 32 bits du pilote ODBC et de l'environnement Java.	

Scénario associé

Pour un scénario associé au **tAccessRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMysqlRollback**.

tAccessRow

Propriétés du tAccessRow

Famille de composant	Databases/Access	
Fonction	**tAccessRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tAccessRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version d'Access que vous utilisez.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à traiter.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Si vous utilisez la version 64 bits du Studio, vous devez utiliser la version 32 bits du pilote ODBC et de l'environnement Java.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du **tMySQLRow**.

tAS400Close

Propriétés du tAS400Close

Famille de composant	Databases/AS/400	
Fonction	Le composant **tAS400Close** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données AS/400.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tAS400Connection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tAS400Connection** et **tAS400Rollback**. Il est généralement utilisé avec un composant **tAS400Connection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant, consultez *tMysqlConnection* du composant **tAS400Close**.

tAS400Commit

Propriétés du tAS400Commit

Famille de composant	Databases/AS/400	
Fonction	Le composant **tAS400Commit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tAS400Connection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tAS400Commit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants AS/400 et notamment avec les composants **tAS400Connection** et **tAS400Rollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tAS400Commit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tAS400Connection

Propriétés du tAS400Connection

Famille de composant	Databases/AS/400	
Fonction	Le composant **tAS400Connection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez la version d'AS/400 que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Auto Commit	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications

		doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec des composants AS/400 et notamment avec les composants **tAS400Commit** et **tAS400Rollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tAS400Connection**, consultez *tMysqlConnection*.

tAS400Input

Propriétés du tAS400Input

Famille de composant	Databases/AS/400	
Fonction	Le composant **tAS400Input** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tAS400SInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Property type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version d'AS/400 que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.

		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type et Query	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé, consultez les scénarios du composant **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tAS400LastInsertId

Propriétés du tAS400LastInsertId

Famille de composant	Databases/AS/400	
Fonction	Le **tAS400LastInsertId** affiche les derniers ID ajoutés à une table à partir d'une connexion AS/400 spécifiée.	
Objectif	Le composant **tAS400LastInsertId** permet de récupérer les clés primaires des entrées dernièrement ajoutées par un utilisateur à une table AS/400.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Component list	Sélectionnez le composant **tAS400Connection** dans la liste s'il y a plus d'une connexion dans votre Job.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé, consultez *Scénario : Récupérer les ID des dernières entrées ajoutées* du composant **tMysqlLastInsertId**.

tAS400Output

Propriétés du tAS400Output

Famille de composant	Databases/AS/400	
Fonction	Le composant **tAS400Output** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tAS400Output** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version d'AS/400 que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore.

		Create table if doesn't exist : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Use commit control*	Cochez cette case pour accéder au champ **Commit every** et définir les opérations de commit. **Commit every** : nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de transaction (cependant pas de rollback) et surtout une meilleure performance des exécutions.

	Additional JDBC parameters	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tAS400Output** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	Use batch size	Cochez cette case pour activer le mode Batch (par lots) pour le traitement des données. Dans le champ **Batch Size**, saisissez le nombre de lignes à traiter dans chaque lot. Cette case est disponible uniquement lorsque vous avez sélectionné l'option **Insert**, **Update** ou **Delete** dans le champ **Action on data**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données AS/400. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tAS400Rollback

Propriétés du tAS400Rollback

Famille de composant	Databases/AS/400	
Fonction	Le **tAS400Rollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tAS400Connection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en association avec des composants AS/400, notamment avec le **tAS400Connection** et le **tAS400Commit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tAS400Rollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMySQLRollback**.

tAS400Row

Propriétés du tAS400Row

Famille de composant	Databases/ AS/400	
Fonction	Le **tAS400Row** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tAS400Row** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Property type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez la version d'AS/400 que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à traiter.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC Parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log lors du traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.	

Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tDB2BulkExec

Propriétés du tDB2BulkExec

Famille de composant	Databases/DB2	
Fonction	Le composant **tDB2BulkExec** exécute des opérations d'Insert sur les données fournies.	
Objectif	Le **tDB2BulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données DB2.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification sur l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.

		Clear a table : supprimer le contenu de la table.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Data file	Nom du fichier à charger. *Le fichier est situé sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations Update et Delete. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (**Advanced settings**) pour une utilisation optimisée de ces opérations.*
Advanced settings	*Additional JDBC parameters*	Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.
	Field terminated by	Caractère, chaîne ou expression régulière séparant les champs.
	Date Format	Utilisez ce champ pour définir l'ordre des mois et des jours.
	Time Format	Utilisez ce champ pour définir l'ordre des heures, minutes et secondes.
	Timestamp Format	Utilisez ce champ pour définir l'ordre de la date et de l'heure.
	Remove Load Pending	Quand la case est cochée, débloque la table en état "Pending", après un chargement massif.
	Load options	Cliquez sur le bouton **[+]** afin d'ajouter des options de chargement des données. **Parameter** : dans la liste, sélectionnez un paramètre de chargement.

		Value : saisissez une valeur pour le paramètre sélectionné.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes DB2.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au composant **tDB2BulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tDB2Close

Propriétés du tDB2Close

Famille de composant	Databases/DB2	
Fonction	Le composant **tDB2Close** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données DB2.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tDB2Connection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tDB2Connection** et **tDB2Rollback**. Il est généralement utilisé avec un composant **tDB2Connection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tDB2Close**, consultez *tMysqlConnection*.

tDB2Commit

Propriétés du tDB2Commit

Famille de composant	Databases/DB2	
Fonction	Le composant **tDB2Commit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tDB2Connection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tDB2Commit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants DB2, notamment le **tDB2Connection** et le **tDB2Rollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tDB2Commit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tDB2Connection

Propriétés du tDB2Connection

Famille de composant	Databases/DB2	
Fonction	Le composant **tDB2Connection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Additional JDBC parameters*	Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.
	Auto Commit	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

		Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec des composants DB2, notamment les composants **tDB2Commit** et **tDB2Rollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé au composant **tDB2Connection**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tDB2Input

Propriétés du tDB2Input

Famille de composant	Databases/DB2	
Fonction	Le composant **tDB2Input** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tDB2Input** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row). Si des guillemets doubles existent dans le nom des colonnes d'une table, les guillemets doubles ne peuvent être récupérés avec les colonnes. Il est donc recommandé de ne pas utiliser de guillemet double dans le nom des colonnes dans les tables de la base de données DB2.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur d'annuaire DB2.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.

		Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table source dans laquelle capturer les changements apportés aux données.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Additional JDBC parameters*	Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données DB2.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez les scénarios du composant **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tDB2Output

Propriétés du tDB2Output

Famille de composant	Databases/DB2	
Fonction	Le composant **tDB2Output** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tDB2Output** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Property type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Action on table	La liste **Action on table** n'est pas disponible si vous cochez la case **Enable parallel execution** dans la vue **Advanced settings**. Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore.

		Create table if doesn't exist : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer le contenu de la table. Il n'y a pas de possibilité de rollback. **Truncate table with reuse storage** : supprimer le contenu de la table. Il n'y a pas de possibilité de rollback. Cependant, vous pouvez réutiliser le stockage alloué à la table, même si le stockage est considéré comme vide. *Si vous cochez la case **Use an existing connection** et que vous sélectionnez **Truncate table** ou **Truncate table with reuse storage** dans la liste **Action on table**, une instruction de commit est invoquée avant l'opération TRUNCATE, car l'instruction TRUNCATE doit être la première instruction dans une transaction.*
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si

		vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de transaction (cependant pas de rollback) et surtout une meilleure performance des exécutions.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tDB2Output** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Convert columns and table names to uppercase	Cochez cette case pour mettre le nom des colonnes et celui de la table en majuscules.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Support null in "SQL WHERE" statement	Cochez cette case pour prendre en compte les valeurs Null d'une table de base de données. Assurez-vous que la case **Nullable** est bien cochée pour les colonnes du schéma correspondantes.
	Use Batch size	Cochez cette case afin d'activer le mode batch pour le traitement des données. Dans le champ **Batch Size** qui apparaît, vous pouvez saisir le nombre de lignes par lot à charger. Cette case est disponible uniquement si vous avez choisi l'option **Insert**, **Update**, **Single Insert Query** ou **Delete** dans le champ **Action on data**.
	tStat Catcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier.	

	NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données DB2. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé au composant **tDB2Output**, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du **tMysqlOutput**.

tDB2Rollback

Propriétés du tDB2Rollback

Famille de composant	Databases/DB2	
Fonction	Le **tDB2Rollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tDB2Connection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en association avec des composants DB2, notamment avec le **tDB2Connection** et le **tDB2Commit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tDB2Rollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMySQLRollback**.

tDB2Row

Propriétés du tDB2Row

Famille de composant	Databases/ DB2	
Fonction	**tDB2Row** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tDB2Row** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Property type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à traiter.
	Query type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**.	

	Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé au composant **tDB2Row**, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tDB2SCD

Le composant **tDB2SCD** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tDB2SCD*.

tDB2SCDELT

Le composant **tDB2SCDELT** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tDB2SCDELT*.

tDB2SP

Propriétés du tDB2SP

Famille de composant	Databases/DB2	
Fonction	Le composant **tDB2SP** appelle une procédure stockée de base de données.	
Objectif	Le **tDB2SP** permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Property type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier des propriétés du composant. Les champs suivants sont alors pré-remplis.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification sur l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il est réutilisable. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	SP Name	Saisissez le nom exact de la Procédure Stockée (SP).
	Is Function / Return result in	Cochez cette case, si une seule valeur doit être retournée. Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.
	Parameters	Cliquez sur le bouton **[+]** et sélectionnez dans le champ **Schema Columns** les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure. Sélectionnez le **Type** de paramètre : **IN** : paramètre d'entrée (Input) **OUT** : paramètre de sortie (Output)/valeur retournée **IN OUT** : les paramètres d'entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction). **RECORDSET** : les paramètres d'entrée doivent être retournées sous forme d'ensemble de valeurs, au lieu d'une valeur unique. Consultez *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.
Advanced settings	*Additional JDBC parameters*	Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d'entrée sont autorisés.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez

- *Scénario : Récupérer des informations personnelles à l'aide d'une procédure stockée*.
- *Scénario : Trouver le libellé State à l'aide d'une procédure stockée*.
- *Scénario : Vérifier le format de numéros à l'aide d'une procédure stockée*.
- *Scénario : Exécuter une procédure stockée dans le hub MDM*.

Consultez également *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.

tInformixBulkExec

Propriétés du tInformixBulkExec

Les composants **tInformixOutputBulk** et **tInformixBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tInformixOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Famille de composant	Databases/Informix	
Fonction	Le composant **tInformixBulkExec** exécute des opérations d'Insert sur les données fournies.	
Objectif	Le **tInformixBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Informix.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Execution Platform	Sélectionnez le type de système d'exploitation que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification sur l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Instance	Nom de l'instance Informix à utiliser. Cette information se trouve généralement dans le fichier *SQL hosts*.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :

		None : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Informix Directory	Répertoire d'installation d'Informix, par exemple *" C:\Program Files\IBM\IBM Informix Dynamic Server\11.50\"*.
	Data file	Nom du fichier à charger.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations Update et Delete. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (**Advanced settings**) pour une utilisation optimisée de ces opérations.*

Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Field terminated by	Caractère, chaîne ou expression régulière séparant les champs.
	Set DBMONEY	Quand vous cochez cette case, vous pouvez définir votre séparateur décimal dans le champ **Decimal separator**.
	Set DBDATE	Sélectionnez le format de date que vous souhaitez appliquer.
	Rows Before Commit	Saisissez le nombre de lignes à traiter avant commit.
	Bad Rows Before Abort	Saisissez le nombre de lignes d'erreur avant arrêt du Job.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Output	Sélectionnez l'emplacement de la sortie.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes DB2.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Le serveur/client de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tInformixBulkExec**, afin que le Job fonctionne correctement. Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au composant **tInformixBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tInformixClose

Propriétés du tInformixClose

Famille de composant	Databases/Informix	
Fonction	Le composant **tInformixClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Informix.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tInformixConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tInformixConnection** et **tInformixRollback**. Ils sont généralement utilisés avec un composant **tInformixConnection** car il permet d'ouvrir une connexion pour la transaction en cours.

Pour un scénario associé au composant **tInformixClose**, consultez *tMysqlConnection*.

tInformixCommit

Propriétés du tInformixCommit

Ce composant est étroitement lié aux composants **tInformixConnection** et **tInformixRollback**. Ils sont généralement utilisés ensemble lors de transactions.

Famille de composant	Databases/Informix	
Fonction	Le composant **tInformixCommit** valide les données traitées dans un Job à partir d'une base de données connectée. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tInformixConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tInformixCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Informix et notamment **tInformixConnection** et **tInformixRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tInformixConnection** et **tInformixRollback**. Ils sont généralement utilisés avec un composant **tInformixConnection** car il permet d'ouvrir une connexion pour la transaction en cours.

Pour un scénario associé au composant **tInformixCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tInformixConnection

Propriétés du tInformixConnection

Ce composant est étroitement lié aux composants **tInformixCommit** et **tInformixRollback**. Ils sont généralement utilisés avec un composant **tInformixConnection** car il permet d'ouvrir une connexion pour la transaction en cours.

Famille de composant	Databases/Informix	
Fonction	Le composant **tInformixConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Instance	Nom de l'instance Informix à utiliser. Cette information se trouve généralement dans le fichier *SQL hosts*.
	Additional JDBC parameters	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Use Transaction*	Décochez cette case lorsque la base de données est configurée en mode NO_LOG. Si la case est cochée, vous pouvez choisir d'activer ou non l'option Auto Commit.

	Auto Commit	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec des composants Informix, notamment les composants **tInformixCommit** et **tInformixRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé au composant **tInformixConnection**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tInformixInput

Propriétés du tInformixInput

Famille de composant	Databases/Informix	
Fonction	Le composant **tInformixInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tInformixInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant adéquat à partir de la liste **Component** pour réutiliser les paramètres d'une connexion que vous avez déjà définis.
	Host name	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Instance	Nom de l'instance Informix à utiliser. Cette information se trouve généralement dans le fichier *SQL hosts*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Guess schema	Cliquez sur le bouton pour récupérer le schéma de la table.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Informix.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Consultez les scénarios associés aux composants **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tInformixOutput

Propriétés du tInformixOutput

Famille de composant	Databases/ Informix	
Fonction	Le composant **tInformixOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tInformixOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Instance	Nom de l'instance Informix à utiliser. Cette information se trouve généralement dans le fichier *SQL hosts*.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle.

		Create a table : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : tronquer la table. *Une opération de commit est effectuée après troncature de la table.*
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tInformixOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	Use Batch Size	Nombre de lignes dans chaque lot de traitement de données.
	Optimize the batch insertion	Cocher cette case optimise l'insertion de données par lots.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Informix. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé au composant **tInformixOutput**, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tInformixOutputBulk

Propriétés du tInformixOutputBulk

Les composants **tInformixOutputBulk** et **tInformixBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tInformixOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/Informix	
Fonction	Ecrit un fichier composé de colonnes et basé sur le délimiteur défini et sur les standards Informix.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Informix.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Row separator*	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Set DBMONEY	Quand vous cochez cette case, vous pouvez définir votre séparateur décimal dans le champ **Decimal separator**.
	Set DBDATE	Sélectionnez le format de date que vous souhaitez appliquer.
	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Custom the flush buffer size	Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ **Row number**, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tInformixBulkExec**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données Informix.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé au **tInformixOutputBulk**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tInformixOutputBulkExec

Propriétés du tInformixOutputBulkExec

Les composants **tInformixOutputBulk** et **tInformixBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tInformixOutputBulkExec**.

Famille de composant	Databases/ Informix	
Fonction	Le **tInformixOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tInformixOutputBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Informix.	
Basic settings	*Property Type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Execution platform	Sélectionnez le type de système d'exploitation que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Instance	Nom de l'instance Informix à utiliser. Cette information se trouve généralement dans le fichier *SQL hosts*.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :

		None : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Clear table** : supprimer le contenu de la table.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Informix Directory	Répertoire d'installation d'Informix, par exemple *" C:\Program Files\IBM\IBM Informix Dynamic Server\11.50\"*.
	Data file	Nom du fichier à générer et à charger.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Action on data	Sélectionnez l'opération que vous voulez effectuer : **Bulk insert** ou **Bulk update**. Les informations demandées seront différentes en fonction de l'action choisie.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs.
	Set DBMONEY	Quand vous cochez cette case, vous pouvez définir votre séparateur décimal dans le champ **Decimal separator**.
	Set DBDATE	Sélectionnez le format de date que vous souhaitez appliquer.
	Rows Before Commit	Saisissez le nombre de lignes à traiter avant commit.
	Bad Rows Before Abort	Saisissez le nombre de lignes d'erreur avant arrêt du Job.

	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Custom the flush buffer size	Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ **Row number**, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Output	Sélectionnez l'emplacement de la sortie.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	
Limitation	Le serveur/client de la base de données doit être installé sur la même machine que le *Studio Talend* ou le Job contenant un **tInformixOutputBulkExec**, afin que le composant fonctionne correctement.	

Scénarios associés

Pour un scénario associé au **tInformixOutputBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tInformixRollback

Propriétés du tInformixRollback

Ce composant est étroitement lié aux composants **tInformixCommit** et **tInformixConnection**. Ils sont généralement utilisés ensemble lors de transactions.

Famille de composant	Databases/Informix	
Fonction	Le **tInformixRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tInformixConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Il faut utiliser ce composant en association avec des composants Informix, notamment avec le **tInformixConnection** et le **tInformixCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tInformixRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMysqlRollback**.

tInformixRow

Propriétés du tInformixRow

Famille de composant	Databases/ Informix	
Fonction	**tInformixRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tInformixRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Instance	Nom de l'instance Informix à utiliser. Cette information se trouve généralement dans le fichier *SQL hosts*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
	Use Transaction	Décochez cette case lorsque la base de données est configurée en mode NO_LOG.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez	

	dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tInformixSCD

Le composant **tInformixSCD** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tInformixSCD*.

tInformixSP

Propriétés du tInformixSP

Famille de composant	Databases/ Informix	
Fonction	Le composant **tInformixSP** appelle une procédure stockée de base de données.	
Objectif	Le **tInformixSP** permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Instance	Nom de l'instance Informix à utiliser. Cette information se trouve généralement dans le fichier *SQL hosts*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	SP Name	Saisissez le nom exact de la Procédure Stockée (SP).
	Is Function / Return result in	Cochez cette case si une seule valeur doit être retournée. Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.
	Parameters	Cliquez sur le bouton **[+]** et sélectionnez dans le champ **Schema Columns** les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure. Sélectionnez le **Type** de paramètre : **IN** : paramètre d'entrée (Input). **OUT** : paramètre de sortie (Output)/valeur retournée. **IN OUT** : les paramètres d'entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction). **RECORDSET** : les paramètres d'entrée doivent être retournées sous forme d'ensemble de valeurs, au lieu d'une valeur unique. Consultez *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.
	Use Transaction	Décochez cette case lorsque la base de données est configurée en mode NO_LOG.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d'entrée sont autorisés.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données. Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Récupérer des informations personnelles à l'aide d'une procédure stockée*.
- *Scénario : Trouver le libellé State à l'aide d'une procédure stockée*.
- *Scénario : Vérifier le format de numéros à l'aide d'une procédure stockée*.
- *Scénario : Exécuter une procédure stockée dans le hub MDM*.

Consultez également *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.

tMSSqlBulkExec

Propriétés du tMSSqlBulkExec

Les composants **tMSSqlOutputBulk** et **tMSSqlBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tMSSqlOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Famille de composant	Databases/MSSql	
Fonction	Effectue une action d'Insert sur les données fournies.	
Objectif	Le composant **tMSSqlBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données MSSql.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.

		Clear a table : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Remote File Name	Nom du fichier à charger. *Le fichier est situé sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
Advanced settings	*Action*	Sélectionnez l'opération que vous voulez effectuer : **Bulk insert** ou **Bulk update Bcp query out**. Les informations demandées seront différentes en fonction de l'action choisie.
Bulk insert & Bulk update	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Fields terminated	Caractère, chaîne ou expression régulière séparant les champs.
	Rows terminated	Caractère, chaîne ou expression régulière séparant les lignes.
	First row	Saisissez le numéro du rang à partir duquel vous voulez démarrer l'action.
	Code page	Cette valeur peut prendre plusieurs formes : **OEM** (valeur par défaut) **ACP RAW** : Personnalisée par l'utilisateur.
	Data file type	Sélectionnez le type de données à traiter.
	Output	Sélectionnez la sortie à utiliser pour récupérer la sortie standard de la base MSSql :

		to console : vers la console. **to global variable** : vers la variable globale.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Bcp query out	*Fields terminated*	Caractère, chaîne ou expression régulière séparant les champs.
	Rows terminated	Caractère, chaîne ou expression régulière séparant les lignes.
	Data file type	Sélectionnez le type de données à traiter.
	Output	Sélectionnez la sortie à utiliser pour récupérer la sortie standard de la base MSSql : **to console** : vers la console. **to global variable** : vers la variable globale.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec un composant **tMSSqlOutputBulk**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données MSSql.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé au composant **tMSSqlBulkExec**, consultez les scénarios suivants :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tMSSqlColumnList

Propriétés du tMSSqlColumnList

Famille de composant	Databases/MS SQL	
Fonction	Le composant **tMSSqlColumList** effectue une opération d'itération dans toutes les colonnes d'une table donnée, grâce à une connexion MS SQL définie.	
Objectif	Le **tMSSqlColumList** permet de faire une liste des libellés des colonnes d'une table MS SQL donnée.	
Basic settings	*Component list*	Sélectionnez le composant **tMSSqlConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Table name	Saisissez le nom de la table.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**COLUMN_NAME** : nom de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **COLUMN_DEFAULT** : valeur par défaut de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **IS_NULLABLE** : nullabilité de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne **YES** si la colonne autorise les valeurs NULL. Sinon, elle retourne **NO**. **DATA_TYPE** : type de données de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CHARACTER_MAXIMUM_LENGTH** : nombre maximum de caractères de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques pour les types de données binaires ou caractères et retourne NULL pour les autres types. **CHARACTER_OCTET_LENGTH** : longueur maximale en octets de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques pour les types de données binaires ou caractères et retourne NULL pour les autres types. **NUMERIC_PRECISION** : précision des données de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques pour les données numériques approximatives, exactes, les entiers, les données monétaires et retourne NULL pour les autres types de données. **NUMERIC_PRECISION_RADIX** : base (radix) de précision des données de la colonne sur laquelle s'effectue l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques pour les données numériques approximatives, exactes, les entiers, les données monétaires et retourne NULL pour les autres types de données. **NUMERIC_SCALE** : l'échelle en chiffres autorisée sur la coloone sur laquelle l'itération s'effectue. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques	

	pour les données numériques approximatives, exactes, les entiers, les données monétaires et retourne NULL pour les autres types de données. **DATETIME_PRECISION** : précision de l'heure et de la date en secondes fractionnelles de la colonne sur laquelle s'effectue l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques si le type de données est *datetime* ou *smalldatetime*. Sinon, retourne NULL. **COLUMN_KEY** : indicateur clé de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères si la colonne est configurée à Key. Sinon, elle ne retourne rien. **NB_COLUMN** : nombre de colonnes itérées jusqu'à présent. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Il faut utiliser ce composant en association avec les autres composants MS SQL, notamment avec le **tMSSqlConnection**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé, consultez *Scénario : Itérer une table de base de données et lister le nom des colonnes de la table*.

tMSSqlClose

Propriétés du tMSSqlClose

Famille de composant	Databases/MSSql	
Fonction	Le composant **tMSSqlClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données MSSql.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tMSSqlConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tMSSqlConnection** et **tMSSqlRollback**. Il est généralement utilisé avec un composant **tMSSqlConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tMSSqlClose**, consultez *tMysqlConnection*.

tMSSqlCommit

Propriétés du tMSSqlCommit

Famille de composant	Databases/MSSql	
Fonction	Le composant **tMSSqlCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tMSSqlConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tMSSqlCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants MSSql et notamment avec les composants **tMSSqlConnection** et **tMSSqlRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tMSSqlCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tMSSqlConnection

Propriétés du tMSSqlConnection

Famille de composant	Databases/MSSQL	
Fonction	Le composant **tMSSqlConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Schema	Nom du schéma.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Additional JDBC parameters	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

		Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants MSSql, notamment les composants **tMSSqlCommit** et **tMSSqlRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Insérer des données dans une table de base de données et en extraire des informations

Ce scénario décrit un Job lisant des données d'un fichier texte, relatives aux employés, insérant les données dans une table d'une base de données MSSQL et extrayant les données les plus utiles de la table, pour les afficher dans la console.

Ce scénario comprend les composants suivants :

- un **tMSSqlConnection** : établit une connexion au serveur MSSQL.
- un **tFileInputDelimited** : lit le fichier d'entrée, définit la structure des données et les envoie au composant suivant.
- un **tMSSqlOutput** : écrit les données qu'il reçoit du composant précédent dans une table d'une base de données MSSQL.
- un **tMSSqlInput** : extrait les données de la table selon une requête SQL.
- un **tLogRow** : affiche les informations reçues du composant précédent dans la console.
- un **tMSSqlCommit** : commite la transaction sur le serveur MSSQL connecté.

Construire le Job

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tMSSqlConnection**, un **tFileInputDelimited**, un **tMSSqlOutput**, un **tMSSqlInput**, un **tLogRow** et un **tMSSqlCommit**.
2. Connectez le **tMSSqlConnection** au **tFileInputDelimited** à l'aide d'un lien **Trigger** > **OnSubjobOk**.
3. Reliez de la même manière le **tFileInputDelimited** au **tMSSqlInput** et le composant **tMSSqlInput** au **tMSSqlCommit**.
4. Connectez le **tFileInputDelimited** au **tMSSqlOutput** à l'aide d'un lien **Row** > **Main**.

5. Répétez l'opération pour connecter le **tMSSqlInput** au **tLogRow**.

Configurer les composants

Ouvrir une connexion au serveur MSSQL

1. Double-cliquez sur le composant **tMSSqlConnection** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **Host**, saisissez l'adresse IP ou le nom de l'hôte du serveur MSSQL, *192.168.30.47* dans cet exemple.
3. Dans le champ **Port**, saisissez le numéro du port du serveur de la base de données, *1433* dans ce scénario.
4. Dans le champ **Schema**, saisissez le nom du schéma, *dbo* dans cet exemple.
5. Dans le champ **Database**, saisissez le nom de la base de données, *talend* dans ce scénario.
6. Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant et votre mot de passe de connexion à MSSQL.

Lire les données d'entrée

1. Double-cliquez sur le composant **tFileInputDelimited** pour ouvrir sa vue **Component**.

2. Cliquez sur le bouton **[...]** à côté du champ **File Name/Stream** afin de parcourir votre système jusqu'au fichier d'entrée, par exemple *D:/Input/Employee_Wage.txt*. Ce fichier texte contient trois colonnes : *id*, *name* et *wage*.

```
id;name;wage
51;Harry;2300
40;Ronald;3796
17;Theodore;2174
21;James;1986
2;George;2591
89;Calvin;2362
84;Ulysses;3383
```

```
4;Lyndon;2264
17;Franklin;1780
86;Lyndon;3999
```

3. Dans le champ **Header**, saisissez *1* pour ignorer la première ligne du fichier d'entrée.

4. Cliquez sur **Edit schema** pour définir les données à passer au composant **tMSSqlOutput**. Dans cet exemple, définissez *id* comme la clé (cochez la case **Key**) et spécifiez la longueur (**Length**) et la précision (**Precision**) pour chaque colonne.

   Cliquez sur **OK** pour fermer l'éditeur de schéma. Une boîte de dialogue s'ouvre, dans laquelle vous pouvez choisir de propager le schéma au composant suivant.

   Pour plus d'informations, consultez *tFileInputDelimited*.

Ecrire les données dans la table de la base de données

1. Double-cliquez sur le **tMSSqlOutput** pour ouvrir sa vue **Basic settings**.

2. Saisissez les informations requises pour la connexion ou utilisez une connexion existante précédemment configurée. Dans cet exemple, cochez la case **Use an existing connection**. Si plusieurs connexions sont disponibles, sélectionnez celle que vous souhaitez utiliser, dans la liste déroulante **Component List**.

3. Dans le champ **Table**, saisissez le nom de la table dans laquelle vous souhaitez écrire les données : *Wage_Info* dans ce scénario. Vous pouvez également cliquer sur le bouton **[...]** à côté du champ **Table** pour ouvrir une boîte de dialogue et sélectionner une autre table.

4. Sélectionnez **Create table if not exists** dans la liste déroulante **Action on table**.

5. Sélectionnez **Insert if not exists** dans la liste déroulante **Action on data**.

6. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.

Extraire des informations de la table

1. Double-cliquez sur le composant **tMSSqlInput** pour afficher sa vue **Component**.

2. Cochez la case **Use an existing connection**. Si plusieurs connexions sont disponibles, sélectionnez celle que vous souhaitez utiliser, dans la liste déroulante **Component List**.

3. Cliquez sur **Edit schema** pour définir la structure des données à lire depuis la table. Dans cet exemple, vous devez lire les trois colonnes de la table.

4. Dans le champ **Table Name**, saisissez le nom de la table de laquelle vous souhaitez lire les données : *Wage_Info* dans ce scénario.

5. Dans le champ **Query**, renseignez la requête SQL à exécuter sur la table spécifiée. Pour obtenir les données relatives aux employés dont le salaire est supérieur à la moyenne et les classer par ID, saisissez la requête SQL comme suit :

```
SELECT    *
FROM      Wage_Info
WHERE     wage >
(SELECT   avg(wage)
FROM      Wage_Info)
ORDER BY  id
```

Afficher les informations dans la console

1. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.
2. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)**.

Commiter la transaction et fermer la connexion

1. Double-cliquez sur le **tMSSqlCommit** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Close Connection**.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.
2. Exécutez le Job en appuyant sur la touche **F6** ou en cliquant sur le bouton **Run** de la vue **Run**.

   La console affiche l'ID des employés dont le salaire est supérieur à la moyenne, leur nom et leur salaire. Les employés sont classés par ID.

tMSSqlInput

Propriétés du tMSSqlInput

Famille de composant	Databases/MSSql	
Fonction	Le composant **tMSSqlInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tMSSqlInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Property type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table name	Nom de la table à lire.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données MS SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez les scénarios des composants **tDBInput** dans :

- *tMSSqlConnection*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du **tContextLoad**.

tMSSqlLastInsertId

Propriétés du tMSSqlLastInsertId

Famille de composant	Databases/MS SQL server	
Fonction	**tMSSqlLastInsertId** affiche les derniers ID ajoutés à une table à partir d'une connexion MSSql spécifiée.	
Objectif	**tMSSqlLastInsertId** permet de récupérer les clés primaires des entrées dernièrement ajoutées par un utilisateur à une table MSSql.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Component list	Sélectionnez le composant **tMSSqlConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé au composant **tMSSqlLastInsertId**, consultez *Scénario : Récupérer les ID des dernières entrées ajoutées*.

tMSSqlOutput

Propriétés du tMSSqlOutput

Famille de composant	Databases/MSSql	
Fonction	Le composant **tMSSqlOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tMSSqlOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Schema	Nom du schéma.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.

	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **Default** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Turn on identity insert	Cochez cette case pour utiliser votre propre séquence sur les valeurs Identity des données insérées (plutôt que de laisser le serveur SQL choisir les valeurs séquentielles).
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Single Insert Query** : Ajouter de nouvelles entrées à la table, regroupées dans un lot. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. **Insert if not exist** : Ajouter de nouvelles entrées à la table si nécessaire. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Specify identity field	Cochez cette case pour spécifier quel est le champ identity (**Identity field**), constitué d'un numéro d'identification incrémenté automatiquement. Lorsque cette case est cochée, trois autres champs s'affichent :

		Identity field : sélectionnez dans la liste la colonne que vous souhaitez définir comme champ identity. **Start value** : saisissez une valeur de départ, utilisée pour la première ligne chargée dans la table. **Step** : saisissez une valeur d'incrément, ajoutée à la valeur de la ligne précédemment chargée. Cette case est disponible si vous sélectionnez l'option **Drop and create table**, **Create table**, **Create table if not exists** ou **Drop table if exists and create** dans la liste **Action on table**. Si vous cochez cette case sans avoir coché la case **Turn on identity insert** mais en ayant sélectionné **Create table if not exists** dans la liste **Action on table** et si la table spécifiée n'existe pas, seule une table est créée sans qu'aucune donnée y soit insérée. Vous pouvez également spécifier le champ identity (**Identity field**) à partir du schéma du composant. Pour ce faire, paramétrez le type de base de données (**DB Type**) de la colonne correspondante en **INT IDENTITY**. Lorsque la case **Specify identity field** est cochée, le type de base de données (**DB Type**) **INT IDENTITY** du schéma est ignoré.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.

Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tMSSqlOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Ignore date validation	Cochez cette case pour ignorer la validation de la date et insérer les données directement dans la base de données pour les types de données DATE, DATETIME, DATETIME2 et DATETIMEOFFSET.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	Support null in "SQL WHERE" statement	Cochez cette case pour prendre en compte les valeurs Null d'une table de base de données. Assurez-vous que la case **Nullable** est bien cochée pour les colonnes du schéma correspondantes.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Use Batch Size	Cochez cette case afin d'activer le mode batch pour le traitement des données. Dans le champ **Batch Size** qui apparaît, vous pouvez saisir le nombre de lignes par lot à charger. Cette case est disponible uniquement si vous avez choisi l'option **Insert**, **Update**, **Single Insert Query** ou **Delete** dans la liste **Action on data**. Si vous sélectionnez l'option **Single Insert Query** dans la liste **Action on data**, la taille des lots doit être inférieure ou égale à la limite des marqueurs de paramètres autorisés par le pilote JDBC (généralement 2000), divisé par le nombre de colonnes.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**.	

	Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données MSSql. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Lorsque l'option **Single Insert Query** est sélectionnée dans la liste **Action on data**, une Instruction Préparée SQL est générée, par exemple, `INSERT INTO table (col1, col2, col3) VALUES (?,?,?) , (?,?,?) , (?,?,?) ,(?,?,?)`. Entre les parenthèses se trouvent les groupes de paramètres, dont le nombre ne peut en général pas dépasser 2000, selon le pilote JBDC. La taille des lots doit être configurée de manière à respecter cette limite, c'est-à-dire qu'elle ne doit pas dépasser 2000. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé au composant **tMSSqlOutput**, consultez :

- *tMSSqlConnection*.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tMSSqlOutputBulk

Propriétés du tMSSqlOutputBulk

Les composants **tMSSqlOutputBulk** et **tMSSqlBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tMSSqlOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/MSSql	
Fonction	Ecrit un fichier composé de colonnes et basé sur le délimiteur défini et sur les standards MSSql.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données MSSQL.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer. *Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.*
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Row separator*	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tMSSQLBulkExec**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données MSSQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé au **tMSSqlOutputBulk**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tMSSqlOutputBulkExec

Propriétés du tMSSqlOutputBulkExec

Les composants **tMSSqlOutputBulk** et **tMSSqlBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tMSSqlOutputBulkExec**.

Famille de composant	Databases/MSSql	
Fonction	Le **tMSSqlOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tMSSqlOutputBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données MSSQL.	
Basic settings	*Action*	Permet de choisir entre les options **Bulk Insert** et **Bulk update**.
	Property Type	Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Host	Adresse IP du serveur de la base de données. Seuls l'adresse IP exacte de la machine locale, *localhost* ou *127.0.0.1* permettent un fonctionnement optimal. Le serveur de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tMSSqlOutputBulkExec**.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table.

		Drop and create the table : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Local File Name	Nom du fichier à générer et à charger. Le fichier est généré sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	First row	Saisissez le numéro du rang à partir duquel vous voulez démarrer l'action.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.

	Data file type	Sélectionnez le type de données à traiter.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	
Limitation	Le serveur de la base de données doit être installé sur la même machine que le *Studio Talend* ou le Job contenant un **tMSSqlOutputBulkExec**, afin que le composant fonctionne correctement.	

Scénarios associés

Pour un scénario associé au **tMSSqlOutputBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tMSSqlRollback

Propriétés du tMSSqlRollback

Famille de composant	Databases	
Fonction	Le **tMSSqlRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tMSSqlConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants MSSql, notamment les composants **tMSSqlConnection** et **tMSSqlCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille*.

tMSSqlRow

Propriétés du tMSSqlRow

Famille de composant	Databases/DB2	
Fonction	**tMSSqlRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tMSSqlRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Property type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit exister pour que l'opération d'Insert soit autorisée.
	Turn on identity insert	Cochez cette case pour utiliser votre propre séquence sur les valeurs Identity des données insérées (plutôt que de laisser le serveur SQL choisir les valeurs séquentielles).
	Query type	La requête peut être **Built-in** ou distante dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.

Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du **tMysqlRow**.

tMSSqlSCD

Le composant **tMSSqlSCD** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tMSSqlSCD*.

tMSSqlSP

Propriétés du tMSSqlSP

Famille de composant	Databases/MSSql	
Fonction	Le composant **tMSSqlSP** appelle une procédure stockée de base de données.	
Objectif	Le **tMSSqlSP** permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier des propriétés du composant. Les champs suivants sont alors pré-remplis.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification sur l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	SP Name	Saisissez le nom exact de la Procédure Stockée (SP).
	Is Function / Return result in	Cochez cette case, si une seule valeur doit être retournée. Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.
	Parameters	Cliquez sur le bouton **[+]** et sélectionnez dans le champ **Schema Columns** les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure. Sélectionnez le **Type** de paramètre : **IN** : paramètre d'entrée (Input) **OUT** : paramètre de sortie (Output)/valeur retournée **IN OUT** : les paramètres d'entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction). **RECORDSET** : les paramètres d'entrée doivent être retournées sous forme d'ensemble de valeurs, au lieu d'une valeur unique. Consultez *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d'entrée sont autorisés.	
Limitation	La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Récupérer des informations personnelles à l'aide d'une procédure stockée

Ce scénario décrit un Job récupérant un enregistrement contenant des informations personnelles à partir d'une table spécifiée. Il utilise pour cela une procédure stockée se basant sur la valeur *id* définie dans le flux d'entrée.

Dans ce scénario, la table de laquelle vous souhaitez récupérer des données contient des informations personnelles. Pour reproduire ce scénario, vous pouvez écrire les données dans une table à partir d'un fichier CSV, comme suit. Pour plus d'informations concernant l'écriture de données dans une table MS SQL, consultez *Scénario : Insérer des données dans une table de base de données et en extraire des informations*.

```
id;name;sex;age
1;Ford;Male;25
2;Rose;Female;30
3;Sabrina;Female;28
4;Teddy;Male;32
5;Kate;Male;35
```

Dans ce scénario, la procédure stockée utilisée pour récupérer les informations personnelles se présente comme suit :

```
CREATE PROCEDURE [dbo].[QueryPerson]
@id int,
@name varchar(50)
AS
BEGIN
SET NOCOUNT ON
SELECT * FROM dbo.person where id=@id
END
```

Construire le Job

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant à partir de la **Palette** : un **tFixedFlowInput**, un **tMSSqlSP**, un **tParseRecordSet** et deux composants **tLogRow**.

2. Reliez le **tFixedFlowInput** au **tMSSqlSP** à l'aide d'un lien **Row** > **Main**.

3. Répétez l'opération pour connecter le **tMSSqlSP** au premier **tLogRow**, le premier **tLogRow** au **tParseRecordSet** et le **tParseRecordSet** au second **tLogRow**.

Configurer les composants

Configurer le composant d'entrée

1. Double-cliquez sur le composant **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

Cliquez deux fois sur le bouton **[+]** pour ajouter deux colonnes : *id* de type **Integer** et *name* de type **String**.

Cliquez sur **OK** pour fermer l'éditeur du schéma.

3. Dans la zone **Mode**, renseignez chaque colonne avec sa valeur respective. Dans cet exemple, la valeur de la colonne *id* est *4* et celle de la colonne *name* est null car elle n'est pas utilisée dans l'instruction SQL SELECT dans la procédure stockée.

Configurer le composant tMSSqlSP

1. Double-cliquez sur le **tMSSqlSP** pour ouvrir sa vue **Basic settings**.

2. Dans les champs **Host**, **Port**, **Schema**, **Database**, **Username** et **Password**, renseignez vos informations de connexion au serveur MS SQL.

3. Dans le champ **SP Name**, saisissez le nom de la procédure stockée à utiliser. Par exemple, saisissez *QueryPerson*.

4. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

Cliquez sur le bouton pour copier toutes les colonnes du schéma d'entrée dans le schéma de sortie. Cliquez sur le bouton **[+]** du panneau de droite afin d'ajouter une nouvelle colonne *person* de type **Object** qui contiendra les informations personnelles à récupérer de la base de données.

Cliquez sur **OK** pour fermer l'éditeur du schéma et acceptez la propagation proposée par la boîte de dialogue.

5. Dans la zone **Parameters**, cliquez trois fois sur le bouton **[+]** pour ajouter trois lignes et sélectionnez une colonne de schéma ainsi que son type pour chaque ligne. Dans cet exemple, les colonnes *id* et *name* sont de type **IN**, et la colonne *person* est de type **RECORD SET**.

Configurer le tParseRecordSet

1. Double-cliquez sur le **tParseRecordSet** pour ouvrir sa vue **Basic settings**.

2. Dans la liste **Prev. Comp. Column list**, sélectionnez la colonne contenant les informations personnelles à parser. Dans cet exemple, sélectionnez *person*.

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

Cliquez quatre fois sur le bouton **[+]** dans le panneau de droite pour ajouter quatre colonnes : *id* et *age* de type **Integer**, et *name* et *sex* de type **String**.

Cliquez sur **OK** pour fermer l'éditeur du schéma et acceptez la propagation proposée par la boîte de dialogue.

4. Dans la table **Attribute table**, renseignez chaque colonne **Value** avec le nom de colonne correspondant dans la table du serveur MS SQL contenant les informations personnelles.

Configurer les composants de sortie

1. Double-cliquez sur le premier **tLogRow** pour ouvrir sa vue **Basic settings**.

Dans la zone **Mode**, sélectionnez **Vertical (each row is a key/value list)** pour un affichage optimal des résultats.

2. Configurez de la même manière le second **tLogRow**.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder votre Job.
2. Appuyez sur **F6** afin de l'exécuter.

```
Starting job tMSSqlSP_Demo at 16:16 29/09/2014.

[statistics] connecting to socket on port 3560
[statistics] connected
.-------------------------------------------------------------.
|                          #1. tLogRow_1                      |
+--------+-----------------------------------------------------+
| key    | value                                               |
+--------+-----------------------------------------------------+
| id     | 4                                                   |
| name   | null                                                |
| person | net.sourceforge.jtds.jdbc.CachedResultSet@22e5b6b0  |
+--------+-----------------------------------------------------+
.--------------.
|#1. tLogRow_2 |
+------+-------+
| key  | value |
+------+-------+
| id   | 4     |
| name | Teddy |
| sex  | Male  |
| age  | 32    |
+------+-------+

[statistics] disconnected
Job tMSSqlSP_Demo ended at 16:16 29/09/2014. [exit code=0]
```

Les informations personnelles contenant l'ID *4* dans la table MS SQL sont affichées dans la table inférieure de la console. Notez que les valeurs des colonnes *id* et *name* affichées dans la table supérieure sont les données d'entrée.

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario : Trouver le libellé State à l'aide d'une procédure stockée*.
- *Scénario : Vérifier le format de numéros à l'aide d'une procédure stockée*.
- *Scénario : Exécuter une procédure stockée dans le hub MDM*.

Consultez également *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.

tMSSqlTableList

Propriétés du tMSSqlTableList

Famille de composant	Databases/MS SQL	
Fonction	Le composant **tMSSqlTableList** effectue une opération d'itération sur toutes les tables d'une base de données, grâce à une connexion MS SQL définie.	
Objectif	Le composant **tMSSqlTableList** permet de lister les noms des tables MS SQL grâce aux commandes SELECT et WHERE.	
Basic settings	*Component list*	Sélectionnez le composant **tMSSqlConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Where clause for table name selection	Saisissez la commande WHERE permettant d'identifier les tables sur lesquelles effectuer l'opération d'itération.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**CURRENT_TABLE** : nom de la table sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_TABLE** : nombre de tables itérées jusqu'à présent. Cette variable est une variable **Flow** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Il faut utiliser ce composant en association avec les autres composants MS SQL, notamment avec le **tMSSqlConnection**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Itérer une table de base de données et lister le nom des colonnes de la table*.

tMysqlBulkExec

Propriétés du tMysqlBulkExec

Les composants **tMysqlOutputBulk** et **tMysqlBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tMysqlOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Famille de composant	Databases/Mysql	
Fonction	Le composant **tMysqlBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tMysqlBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données MySQL.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez la version de MySQL que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune action sur la table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore.

		Create table if doesn't exist : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer un nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Table	Nom de la table à écrire. Une seule table peut être écrite à la fois et cette table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Local file Name	Nom du fichier à charger. Ce fichier doit se situer sur la même machine que le *Studio Talend* ou que le Job contenant un **tMysqlBulkExec**, afin que le composant fonctionne correctement.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.
	Lines terminated by	Caractère ou suite de caractères utilisés pour séparer les lignes.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs.
	Enclosed by	Caractères entourant les données.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert records in table** : Insère de nouveaux enregistrements dans une table. **Replace records in table** : Remplace les enregistrements existants par de nouveaux. **Update records in table** : Modifie les enregistrements existants. **Ignore records in table** : Ignore les enregistrements existants ou insère les nouveaux.
	Records contain NULL value	Cochez cette case si vous souhaitez récupérer les valeurs nulles du flux d'entrée. Si vous ne cochez pas cette case, les valeurs nulles en entrée seront considérées comme des champs vides dans le flux de sortie.

	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec un composant **tMysqlOutputBulk**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données MySQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour plus d'informations relatives au fonctionnement de **tMysqlBulkExec**, consultez les scénarios suivants dans :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du composant **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du composant **tOracleBulkExec**.

tMysqlClose

Propriétés du tMysqlClose

Famille de composant	Databases/Mysql	
Fonction	Le composant **tMysqlClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Mysql.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tMysqlConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être utilisé avec des composants MySQL, notamment avec le **tMysqlCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tMysqlConnection** et **tMysqlRollback**. Il est généralement utilisé avec un composant **tMysqlConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tMysqlClose**, consultez *tMysqlConnection*.

tMysqlColumnList

Propriétés du tMysqlColumnList

Famille de composant	Databases/MySQL	
Fonction	Le composant **tMysqlColumList** effectue une opération d'itération dans toutes les colonnes d'une table donnée, grâce à une connexion MySQL définie.	
Objectif	Le **tMysqlColumList** permet de faire une liste des libellés des colonnes d'une table MySQL donnée.	
Basic settings	*Component list*	Sélectionnez le composant **tMysqlConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Table name	Saisissez le nom de la table.
Global Variables	**COLUMN_NAME** : nom de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **DATA_TYPE** : type de données de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **COLUMN_DEFAULT** : valeur par défaut de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **IS_NULLABLE** : nullabilité de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne **YES** si la colonne autorise les valeurs NULL. Sinon, elle retourne **NO**. **COLUMN_KEY** : indicateur clé de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères si la colonne est configurée à Key. Sinon, elle ne retourne rien. **CHARACTER_MAXIMUM_LENGTH** : nombre maximum de caractères de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques pour les types de données binaires ou caractères et retourne NULL pour les autres types. **NUMERIC_PRECISION** : précision des données de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques pour les données numériques approximatives, exactes, les entiers, les données monétaires et retourne NULL pour les autres types de données. **NUMERIC_SCALE** : l'échelle en chiffres autorisée sur la coloone sur laquelle l'itération s'effectue. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques pour les données numériques approximatives, exactes, les entiers, les données monétaires et retourne NULL pour les autres types de données. **NB_COLUMN** : nombre de colonnes itérées jusqu'à présent. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Il faut utiliser ce composant en association avec les autres composants MySQL, notamment avec le **tMysqlConnection**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Itérer une table de base de données et lister le nom des colonnes de la table

Le scénario Java suivant permet de créer un Job à cinq composants. Ce Job permet d'itérer une table donnée à partir d'une base de données MySQL en utilisant la commande Where et de lister tous les libellés des colonnes de la table.

- Dans la **Palette**, cliquez sur chacun des composants suivants et déposez-les dans l'espace de modélisation : **tMysqlConnection**, **tMysqlTableList**, **tMysqlColumnList**, **tFixedFlowInput** et **tLogRow**.
- Reliez le **tMysqlConnection** au **tMysqlTableList** par le biais d'un lien **OnSubjobOk**.
- Reliez les composants **tMysqlTableList**, **tMysqlColumnList** et **tFixedFlowInput** à l'aide de liens **Iterate**.
- Reliez le **tFixedFlowInput** au **tLogRow** par le biais d'un lien **Row Main**.

- Dans l'espace de modélisation, sélectionnez le composant **tMysqlConnection** et cliquez sur la vue **Component** pour définir les paramètres de base.
- Dans l'onglet **Basic settings**, entrez manuellement les propriétés de connexion à la base de données ou sélectionnez-les à partir de la liste des variables de contexte : appuyez simultanément sur les touches **Ctrl**

+Espace puis cliquez sur le champ correspondant si vous avez stocké les informations en local sous le nœud **DB Connections** du répertoire **Metadata**.

Pour plus d'informations sur les métadonnées, veuillez consulter le *Guide utilisateur du Studio Talend*.

Dans cet exemple, connectez-vous à une base de données MySQL appelée *customers*.

- Dans l'espace de modélisation, sélectionnez le composant **tMysqlTableList** et cliquez sur la vue **Component** pour définir ses paramètres de base.

- Dans le champ **Component list**, sélectionnez le composant de connexion MySQL adéquat si vous utilisez plus d'une connexion.
- Renseignez la clause Where en utilisant la bonne syntaxe dans le champ correspondant pour itérer les noms de la ou des tables que vous voulez lister dans la console.

  Dans ce scénario, itérez la table appelée *customer*.
- Dans l'espace de modélisation, sélectionnez le composant **tMysqlColumnList** et cliquez sur la vue **Component** pour paramétrer ses propriétés de base.

- Dans le champ **Component list**, sélectionnez le composant de connexion MySQL adéquat si vous utilisez plus d'une connexion.
- Dans le champ **Table name**, saisissez le nom de la table dont vous souhaitez lister les libellés de colonnes.

  Dans ce scénario, l'objectif est de lister les libellés des colonnes de la table *customer*.

- Dans l'espace de modélisation, sélectionnez le composant **tFixedFlowInput** et cliquez sur la vue **Component** pour paramétrer ses propriétés de base.
- Dans le champ **Schema**, sélectionnez l'option **Built-In** et cliquez sur le bouton **[...]** à côté du champ **Edit Schema** pour définir les données que vous souhaitez utiliser en entrée. Dans ce scénario, le schéma est composé de deux colonnes : la première *TableName* correspond au nom de la table MySQL et la deuxième *ColumnName* correspond aux libellés des colonnes.

- Cliquez sur **OK** pour fermer la boîte de dialogue et une nouvelle boîte de dialogue apparaît vous demandant si vous souhaitez propager ces modifications, acceptez en cliquant sur **Yes**. Les colonnes définies apparaissent dans le champ **Values** de la vue **Basic settings**.
- Dans le champ **Values**, cliquez sur chacune des cellules de la colonne **Value** pour définir le contenu des colonnes de la table MySQL et appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales.
- Dans la liste des variables globales, sélectionnez : *((String)globalMap.get("tMysqlTableList_1_CURRENT_TABLE"))* pour la colonne *TableName* et

  ((String)globalMap.get("tMysqlColumnList_1_COLUMN_NAME")) pour la colonne *ColumnName*.

- Dans l'espace de modélisation, sélectionnez le composant **tLogRow**.
- Cliquez sur la vue **Component** et paramétrez les propriétés de base du **tLogRow** en fonction de vos besoins.
- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job Column_Table_List at 00:55 17/11/2008.
customer|id
customer|First_Name
customer|Last_Name
customer|Address
customer|id_State
Job Column_Table_List ended at 00:55 17/11/2008. [exit code=0]
```

Le nom de la table MySQL et le libellé de ses colonnes sont affichés dans la console.

tMysqlCommit

Propriétés du tMysqlCommit

Famille de composant	Databases/MySQL	
Fonction	Le composant **tMysqlCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tMysqlConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tMysqlCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants MySQL et notamment avec les composants **tMysqlConnection** et **tMysqlRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé au composant **tMysqlCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tMysqlConnection

Propriétés du tMysqlConnection

Famille de composant	Databases/MySQL	
Fonction	Le composant **tMysqlConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un

		composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants MySQL, notamment les composants **tMysqlCommit** et **tMysqlRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Insérer des données dans des tables mère/fille

Le Job suivant est dédié à des utilisateurs avancés de base de données qui souhaitent effectuer des insertions dans des tables multiples à l'aide de l'*id* d'une table parent qui alimentera une table enfant.

Avant de créer ce Job, suivez les étapes décrites ci-dessous pour créer les tables nécessaires, à l'aide d'un gestionnaire de table tel que *innodb*.

1. Dans un éditeur de ligne de commande, connectez-vous à votre serveur MySQL.
2. Une fois connecté à la base de données correspondante, saisissez la commande suivante afin de créer la table parent :

```
create table f1090_mum(id int not null auto_increment, name varchar(10), primary
 key(id)) engine=innodb;
```

3. Puis créez la deuxième table :

```
create table baby (id_baby int not null, years int) engine=innodb;
```

Dans le *Studio Talend*, le Job est constitué de sept composants, notamment un **tMysqlConnection** et un **tMysqlCommit**.

Construire le Job

1. Cliquez et déposez les composants suivants de la **Palette** : un **tFileList**, un **tFileInputDelimited**, un **tMap**, un **tMysqlConnection**, un **tMysqlCommit** et deux **tMysqlOutput**.
2. Connectez le composant **tMysqlConnection** au **tFileList** à l'aide d'une connexion de type **OnComponentOk**.
3. Connectez le composant **tFileList** au **tMysqlCommit** à l'aide d'un lien **OnComponentOk**.
4. Reliez le **tFileList** au **tFileInputDelimited** à l'aide d'un lien **Iterate**. Le nom du fichier à traiter est automatiquement renseigné depuis le répertoire du **tFileList**, grâce à une variable globale.

5. Reliez le composant **tFileInputDelimited** au **tMap** et divisez le flux entre les deux composants Mysql de sortie. Utilisez un lien **Row** > **Main** pour chaque connexion, représentant le flux de données principal.

Configurer les composants

1. Configurez les propriétés du **tFileList**, notamment le nom du répertoire duquel les fichiers vont être récupérés.

2. Dans la vue **Component** du composant **tMysqlConnection**, définissez les informations de connexion manuellement ou récupérez-les dans le Repository si vous les avez préalablement stockées dans le dossier **DB connection** du répertoire **Metadata**. Pour plus d'informations concernant les Métadonnées, consultez le *Guide utilisateur du Studio Talend*.

3. Dans l'onglet **Basic settings** du composant **tFileInputDelimited**, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. Renseignez le champ **File Name** avec la variable globale : *tFileList_1.CURRENT_FILEPATH*

4. Paramétrez les autres champs comme vous le souhaitez, définissez les séparateurs de lignes et de champs (**Row** et **Field Separator**) en fonction de la structure de votre document.

5. Définissez manuellement le schéma du fichier à l'aide du bouton **Edit schema** ou sélectionnez le schéma dans le **Repository**. En version Java, assurez-vous que le type de données est correctement défini, conformément à la nature des données traitées.

6. Dans la zone Output du **tMap**, ajoutez deux tables de sortie, une que vous nommerez *mum* pour la table parent et une deuxième que vous nommerez *baby* pour la table enfant.

Cliquez sur la colonne *Name* de la zone **Input** et glissez-la dans la table *mum*.

Cliquez sur la colonne *Years* de la zone **Input** et glissez-la dans la table *baby*.

7.

8. Assurez-vous que la table *mum* est bien au dessus de la table *baby* car l'ordre des tables détermine la séquence des flux et ainsi l'insertion de base de données est exécutée correctement.

   Sélectionnez une connexion de sortie de type **Row** pour acheminer correctement le flux vers le composant de sortie correspondant.

9. Dans l'onglet **Basic settings** des deux composants **tMysqlOutput**, cochez la case **Use an existing connection** pour récupérer les informations du composant **tMysqlConnection**.

10. Nommez votre table dans le champ **Table** en vous assurant que vous saisissez un nom pour la bonne table. Dans cet exemple, la table est soit *f1090_mum*, soit *f1090_baby*.

    Sélectionnez **Insert** dans le champ **Action on data** pour les deux composants de sortie (**tMysqlOutput**).

    Cliquez sur le bouton **Sync columns** pour récupérer le schéma défini dans le **tMap**.

11. Dans la zone **Additional columns** de l'onglet **Advanced settings** du composant **tMysqlOutput** correspondant à la table enfant (*f1090_baby*), définissez la colonne *id_baby* de manière à ce que l'*id* de la table parent soit réutilisé.

12. Dans le champ **SQL expression**, saisissez : `'(Select Last_Insert_id())'`

    Dans la colonne **Position**, sélectionnez l'option *Before* et dans **Reference column**, l'option *years*.

    Dans l'onglet **Advanced settings**, décochez la case **Extend insert**.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour enregistrer votre Job.
2. Appuyez sur **F6** pour l'exécuter.

```
mysql> select * from f1090_mum
    -> ;
+----+--------+
| id | names  |
+----+--------+
|  6 | john   |
|  7 | bruce  |
|  8 | beth   |
|  9 | andrew |
| 10 | donald |
| 11 | betty  |
| 12 | john   |
| 13 | bruce  |
| 14 | beth   |
| 15 | andrew |
| 16 | donald |
| 17 | betty  |
+----+--------+
12 rows in set (0.00 sec)
```

```
mysql> select * from f1090_baby
    -> ;
+---------+-------+
| id_baby | years |
+---------+-------+
|       6 |    10 |
|       7 |    23 |
|       8 |    34 |
|       9 |    10 |
|      10 |    23 |
|      11 |    34 |
|      12 |    10 |
|      13 |    23 |
|      14 |    34 |
|      15 |    10 |
|      16 |    23 |
|      17 |    34 |
+---------+-------+
12 rows in set (0.00 sec)
```

L'*id* de la table parent a été réutilisé pour alimenter la colonne *id_baby*.

Scénario 2 : Partager une connexion de base de données entre un Job père et un Job fils

Ce scénario montre comment partager une connexion entre un Job père et un Job fils. Le Job père appelle le Job fils afin d'écrire des données aléatoires dans une base de données MySQL. Il lit ensuite les données depuis la base de données MySQL puis les affiche dans la console. La connexion à la base de données MySQL est configurée une seule fois et utilisée dans les deux Jobs.

Configurer le Job fils

Ajouter et relier les composants

1. Ajoutez les composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette** :
   - un **tMysqlConnection**, afin d'ouvrir une connexion à la base de données MySQL,
   - un **tRowGenerator**, afin de générer des données d'entrée aléatoires,
   - un **tMysqlOutput**, afin d'écrire des données dans la base de données MySQL.
2. Reliez le **tRowGenerator** au **tMysqlOutput** à l'aide d'un lien **Row** > **Main**.
3. Reliez le **tMysqlConnection** au **tRowGenerator** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Configurer la connexion à la base de données

1. Double-cliquez sur le **tMysqlConnection** afin d'ouvrir sa vue **Basic settings**.

2. Dans la liste **Property Type**, sélectionnez **Built-In** puis configurez les détails de connexion dans les champs appropriés, dont :

   - le nom d'hôte ou l'adresse IP de votre serveur de base de données (**Host**),
   - le numéro du port d'écoute (**Port**),
   - le nom de la base de données (**Database**),
   - le nom d'utilisateur (**Username**) et le mot de passe (**Password**) pour l'identification à la base de données.

   Si vous avez stocké les informations de connexion sous le nœud **Metadata** du **Repository**, vous pouvez déposer ces métadonnées centralisées sur le composant **tMysqlConnection**. Pour plus d'informations concernant la centralisation d'une connexion à une base de données, consultez le chapitre sur la gestion des métadonnées dans le *Guide utilisateur du Studio Talend*.

3. Cochez la case **Use or register a shared DB Connection** afin que la connexion à la base de données ouverte par ce composant puisse être partagée entre les différents Jobs. Ensuite, dans le champ **Shared DB Connection Name**, saisissez un nom pour la connexion partagée, entre guillemets doubles, *shared_mysql_connection* dans cet exemple.

Configurer les données d'entrée

1. Double-cliquez sur le composant **tRowGenerator** afin d'ouvrir l'éditeur **Row Generator**.

2. Cliquez sur le bouton **[+]** afin d'ajouter quatre colonnes puis définissez leurs propriétés comme suit :
   - *id*, de type **Integer** et d'une longueur de 2 caractères,
   - *firstName*, de type **String** et d'une longueur de 15 caractères,
   - *lastName*, de type **String** et d'une longueur de 15 caractères,
   - *city*, de type **String** et d'une longueur de 15 caractères.
3. Définissez la fonction de chaque colonne :
   - Pour la colonne *id*, sélectionnez **Numeric.sequence** dans la liste **Function** afin de générer des séquences de nombres.
   - Pour la colonne *firstName*, sélectionnez **TalendDataGenerator.getFirstName** dans la liste **Function** afin de générer des prénoms de manière aléatoire.
   - Pour la colonne *lastName*, sélectionnez **TalendDataGenerator.getLastName** dans la liste **Function** pour générer des noms de famille de manière aléatoire.
   - Pour la colonne *city*, sélectionnez **TalendDataGenerator.getUsCity** dans la liste **Function** pour générer des noms de ville de manière aléatoire.
4. Dans le champ **Number of Rows for RowGenerator**, précisez le nombre de colonnes de données que vous souhaitez générer, *10* dans cet exemple.
5. Cliquez sur le bouton **Preview** de l'onglet **Preview** afin de valider les paramètres du générateur.
6. Une fois les paramètres validés, cliquez sur **OK** afin de fermer l'éditeur. Dans la boîte de dialogue qui s'affiche, cliquez sur **Yes** afin de propager les changements au composant suivant.

Configurer la sortie de la base de données

1. Double-cliquez sur le **tMysqlOutput** afin d'ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** et, si vous avez plus d'un composant de connexion dans ce Job, sélectionnez le composant de connexion à utiliser dans la liste **Component List**.

3. Dans le champ **Table**, saisissez le nom de la table de base de données dans laquelle les données sont écrites, *customers* dans cet exemple.

4. Dans la liste **Action on table**, sélectionnez l'option **Drop table if exists and create** afin de vous assurer de la création d'une table vide.

5. Dans la liste **Action on data**, sélectionnez **Insert**.

6. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin de vérifier la propagation du schéma de sortie. Si besoin, cliquez sur **Sync columns** afin de récupérer le schéma depuis le composant précédent.

7. Appuyez sur **Ctrl+S** afin de sauvegarder votre Job.

Configurer le Job père

Ajouter et relier les composants

1. Ajoutez les composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette** :

   - un **tRunJob**, afin d'appeler le Job fils,
   - un **tMysqlConnection**, afin d'ouvrir la connexion à la base de données MySQL,
   - un **tMysqlInput**, afin de lire les données écrites dans la base de données MySQL par le Job fils,
   - un **tLogRow**, afin d'afficher les données dans la console,
   - un **tMysqlCommit**, afin de commiter les données lors de la transaction et fermer la connexion à la base de données.

2. Reliez le **tRunJob** au **tMysqlConnection** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

3. Reliez le **tMysqlConnection** au **tMysqlInput** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

4. Reliez le **tMysqlInput** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

5. Reliez le **tMysqlInput** au **tMysqlCommit** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Configurer les composants

1. Double-cliquez sur le **tRunJob** afin d'ouvrir sa vue **Basic settings**.

   Si un Job fils a déjà été défini pour le composant, vous pouvez cliquer-droit sur le composant et sélectionner **Settings** dans le menu contextuel afin d'ouvrir la vue **Basic settings**. Vous pouvez également cliquer sur le composant et sélectionner l'onglet **Component**.

2. Cliquez sur le bouton **[...]** à côté du champ **Job** et sélectionnez le Job fils dans la boîte de dialogue **[Repository Content]**. Laissez tous les autres paramètres tels qu'ils sont.

3. Dans la vue **Basic settings** du composant **tMysqlConnection**, cochez la case **Use or register a shared DB Connection**. Ensuite, dans le champ **Shared DB Connection Name**, saisissez le nom de la connexion partagée définie dans le Job fils, *shared_mysql_connection* dans cet exemple.

   Laissez les autres paramètres vides ou tels qu'ils sont.

4. Double-cliquez sur le **tMysqlInput** afin d'ouvrir sa vue **Basic settings**.

5. Cochez la case **Use an existing connection** et, dans le cas où vous avez plus d'un composant de connexion dans ce Job, sélectionnez le composant de connexion à utiliser dans la liste **Component List**.

6. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir l'éditeur **[Schema]** et définissez la même structure de données que dans le Job fils.

Une fois la structure de données définie, cliquez sur **OK** afin de fermer l'éditeur de schéma. Dans la boîte de dialogue qui s'affiche, cliquez sur **Yes** afin de propager les changements au composant suivant.

7. Précisez le nom de la tâche dans la liste **Table Name** et cliquez sur **Guess Query** afin de renseigner automatiquement le champ **Query** avec la requête appropriée.

8. Dans la vue **Basic settings** du **tLogRow**, sélectionnez le mode **Table** afin d'afficher le résultat de l'exécution sous forme de tableau.

   Laissez les autres paramètres du **tMysqlCommit** tels qu'ils sont.

Exécuter le Job

1. Appuyez sur **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la console **Run** afin d'exécuter votre Job.

Le Job père appelle le Job fils afin d'écrire des données dans la base de données. Il lit ensuite les données depuis la base de données puis les affiche dans la console.

tMysqlInput

Propriétés du tMysqlInput

Famille de composant	Databases/MySQL	
Fonction	Le composant **tMysqlInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tMysqlInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.

		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à lire.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. *Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l'alias de la source de données spécifié.* Cette option est indisponible lorsque la case **Use an existing connection** est cochée.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Lorsque vous devez traiter des données au format date/heure *0000-00-00 00:00:00* utilisant ce composant, définissez les paramètres comme suit : `noDatetimeStringSync=true&zero-DateTimeBehavior=convertToNull`.
	Enable stream	Cochez cette case pour déterminer avec quelles lignes vous souhaitez travailler. Cette option permet d'améliorer les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées. Décochez **Trim all the String/Char columns** pour activer le tableau **Trim column**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.	

	QUERY : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données MySQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario 1 : Ecrire des colonnes d'une base de données MySQL dans un fichier de sortie

Dans ce scénario, vous allez lire des colonnes d'une base de données MySQL puis les écrire dans une table d'un fichier local de sortie.

Déposer et relier les composants

1. Déposez un composant **tMysqlInput** et un **tFileOutputDelimited** de la **Palette** dans l'espace de modélisation graphique.

2. Reliez le **tMysqlInput** au **tFileOutputDelimited** à l'aide d'un lien **Row > Main**.

Configurer les composants

1. Double-cliquez sur le composant **tMysqlInput** afin d'ouvrir l'onglet **Basic settings** de la vue **Component**.

2. Dans la liste **Property Type**, sélectionnez **Repository**, si vous avez déjà stocké la connexion à la base de données dans le nœud **Metadata** du **Repository**. Les informations de connexion sont alors automatiquement renseignées.

   Pour plus d'informations concernant le stockage des métadonnées dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

   Si vous n'avez pas stocké localement la connexion dans le **Repository**, sélectionnez **Built-In** dans la liste **Property Type** puis renseignez manuellement les informations.

3. Dans la liste **Schema**, sélectionnez **Built-In** puis cliquez sur **Edit schema** pour configurer le schéma.

   L'éditeur du schéma s'ouvre :

4. Cliquez quatre fois sur le bouton pour ajouter quatre lignes au schéma.

   Dans les colonnes **Column** et **Db Column**, cliquez dans les champs afin de saisir le nom des colonnes correspondantes, *id*, *first_name*, *city* et *salary*, respectivement.

   Cliquez sur les champs de la colonne **Type** pour définir le type de données.

Cliquez sur **OK** pour fermer l'éditeur du schéma.

5. A côté du champ **Table Name**, cliquez sur le bouton **[...]** pour sélectionner la table de la base de données souhaitée.

   Une boîte de dialogue affiche une arborescence de toutes les tables dans la base de données sélectionnée :

6. Cliquez sur la table qui vous intéresse puis sur **OK** pour fermer la boîte de dialogue.

7. Dans la liste **Query Type**, sélectionnez **Built-In**.

8. Dans le champ **Query**, saisissez la requête permettant de récupérer les colonnes souhaitées de la table.

9. Double-cliquez sur le **tFileOutputDelimited** afin d'afficher sa vue **Component**.

10. A côté du champ **File Name**, cliquez sur le bouton **[...]** afin de parcourir votre système jusqu'à l'emplacement où vous souhaitez sauvegarder le fichier de sortie, puis saisissez un nom pour le fichier.

    Cochez la case **Include Header** afin de récupérer le nom des colonnes du schéma ainsi que les données.

11. Sauvegardez le Job en appuyant sur les touches **Ctrl+S**.

Exécuter le Job

Appuyez sur **F6** pour exécuter votre Job.

```
1  id;first_name;city;salary
2  1;Martin;Sacramento;9011
3  2;Zachary;Atlanta;8118
4  3;James;Hartford;5087
5  4;Herbert;Charleston;9233
6  5;Herbert;Hartford;6289
7  6;Ulysses;Nashville;6269
```

Dans le fichier de sortie est écrit le nom des colonnes souhaitées et les données correspondantes, récupérés de la base de données :

Le Job peut également être exécuté en mode **Traces Debug**, ce qui vous permet de voir les lignes pendant leur écriture dans le fichier de sortie, dans l'espace de modélisation graphique.

Scénario 2 : Utiliser des paramètres de contexte lors de la lecture d'une table d'une base de données MySQL

Dans ce scénario, une table d'une base de données MySQL est lue, à l'aide un paramètre de contexte référant au nom de la table.

Déposer et relier les composants

1. Déposez un composant **tMysqlInput** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Reliez le **tMysqlInput** au **tLogRow** à l'aide d'un lien **Row > Main**.

Configurer les composants

1. Double-cliquez sur le **tMysqlInput** pour ouvrir la vue **Basic settings**.

2. Dans la liste **Property Type**, sélectionnez **Repository** si vous avez déjà stocké la connexion à la base de données dans le nœud **Metadata** du **Repository**. Les champs des propriétés sont automatiquement renseignés.

   Pour plus d'informations concernant le stockage d'une connexion à une base de données, consultez le *Guide utilisateur du Studio Talend*.

   Si vous n'avez pas configuré de connexion à une base de données dans le **Repository**, renseignez manuellement les informations après avoir sélectionné **Built-in** dans la liste **Property Type**.

3. Configurez le **Schema** en **Built-In**, puis cliquez sur **Edit schema** pour définir le schéma comme vous le souhaitez.

   L'éditeur du schéma s'ouvre :

4. Cliquez sur le bouton pour ajouter les sept colonnes à utiliser pour configurer le schéma et renommez-les respectivement : *id*, *first_name*, *last_name*, *city*, *state*, *date_of_birth* et *salary*.

   Cliquez sur les lignes de la colonne **Type** pour définir le type des données.

   Cliquez sur **OK** pour fermer l'éditeur du schéma.

5. Placez votre souris sur le champ **Table Name** et appuyez sur **F5** pour configurer les paramètres de contexte.

Pour plus d'informations concernant les paramètres de contexte, consultez le *Guide utilisateur du Studio Talend*.

6. Laissez les paramètres par défaut dans le champ **Name** et saisissez le nom de la table de la base de données dans le champ **Default value**, *employees* dans ce cas.

7. Cliquez sur **Finish** pour valider.

   Le paramètre de contexte *context.TABLE* apparaît automatiquement dans le champ **Table Name**.

8. Dans la liste **Query type**, sélectionnez **Built-In** puis cliquez sur **Guess Query** pour obtenir l'instruction de la requête.

   Dans ce scénario, lisez les enregistrements contenant un salaire supérieur à 8000. Ajoutez une clause `Where`. L'instruction finale s'affiche comme suit :

```
"SELECT
  "+context.TABLE+".`id`,
  "+context.TABLE+".`first_name`,
  "+context.TABLE+".`last_name`,
  "+context.TABLE+".`city`,
  "+context.TABLE+".`state`,
  "+context.TABLE+".`date_of_birth`,
  "+context.TABLE+".`salary`
FROM "+context.TABLE+"
WHERE
  "+context.TABLE+".`salary` > 8000"
```

9. Double-cliquez sur le **tLogRow** pour configurer ses propriétés de base dans l'onglet **Basic settings**.

10. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un meilleur affichage des résultats.

11. Sauvegardez le Job.

Exécuter le Job

Appuyez sur **F6** pour exécuter votre Job. Les résultats s'affichent dans la console.

```
.--+----------+---------+----------+-----+-------------+------.
|                        tLogRow_2                             |
|=-+----------+---------+----------+-----+-------------+-----=|
|id|first_name|last_name|city      |state|date_of_birth|salary|
|=-+----------+---------+----------+-----+-------------+-----=|
|1 |Martin    |Roosevelt|Sacramento|ND   |19-10-1950   |9011  |
|2 |Zachary   |Johnson  |Atlanta   |ND   |14-11-1978   |8118  |
|4 |Herbert   |Harrison |Charleston|AB   |15-08-1963   |9233  |
'--+----------+---------+----------+-----+-------------+------'
```

Comme vous pouvez le constater ci-dessus, les enregistrements comportant un salaire supérieur à 8000 sont récupérés.

Scénario 3 : Lire des données dans des bases de données MySQL à l'aide de connexions dynamiques basées sur les variables de contexte

Dans ce scénario, vous allez lire des données depuis des tables de bases de données ayant la même structure mais se trouvant dans deux différentes bases de données MySQL nommées respectivement *project_q1* et *project_q2*. Vous allez spécifier dynamiquement les connexions à ces bases de données au moment de l'exécution, sans modifier le Job.

Déposer et relier les composants

1. Déposez deux composants **tMysqlConnection**, un **tMysqlInput**, un **tLogRow** et un **tMysqlClose** dans l'espace de modélisation graphique.

2. Reliez le premier **tMysqlConnection** au second **tMysqlConnection** et le second **tMysqlConnection** au **tMysqlInput** à l'aide de liens **Trigger** > **On Subjob Ok**.

3. Reliez le **tMysqlInput** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

4. Reliez le **tMysqlInput** au **tMysqlClose** à l'aide d'un lien **Trigger** > **On Subjob Ok**.

Créer une variable de contexte

Afin de pouvoir choisir dynamiquement une connexion à la base de données au moment de l'exécution, vous devez définir une variable de contexte. Cette dernière est configurée dans les paramètres **Dynamic settings** du composant d'entrée de la base de données.

1. Dans la vue **Contexts**, cliquez sur le bouton **[+]** afin d'ajouter une ligne à la table, puis cliquez dans le champ **Name** et saisissez un nom pour la variable, *myConnection* dans cet exemple.

2. Dans le champ de la liste **Type**, sélectionnez **List Of Value**.

3. Cliquez dans le champ **Value**, puis cliquez sur le bouton qui apparaît dans ce champ pour ouvrir la boîte de dialogue **[Configure value of list]**.

4. Dans la boîte de dialogue **[Configure value of list]**, cliquez sur le bouton **New...** pour ouvrir la boîte de dialogue **[New Value]** et saisissez le nom de l'un des composants de connexion dans le champ de texte, *tMysqlConnection_1* dans cet exemple. Cliquez ensuite sur **OK** pour fermer la boîte de dialogue.

Répétez cette étape pour spécifier le nom de l'autre composant de connexion comme autre élément de liste, *tMysqlConnection_2* dans cet exemple.

Lorsque vous avez terminé, cliquez sur **OK** pour fermer la boîte de dialogue **[Configure Values]**.

5. Cochez la case à côté du champ de valeur de la variable et saisissez, dans le champ **Prompt**, le message que vous souhaitez afficher à l'exécution, *Select a connection component:* dans cet exemple.

Configurer les composants

1. Double-cliquez sur le premier composant **tMysqlConnection** afin d'afficher sa vue **Basic settings** et saisissez les informations de connexion. Pour plus d'informations concernant la configuration du **tMysqlConnection**, consultez *tMysqlConnection*.

   Notez que ce composant sert à ouvrir une connexion à une base de données MySQL nommée *project_q1*.

2. Configurez le second composant **tMysqlConnection** de la même façon mais en saisissant *project_q2* dans le champ **Database**. En effet, ce composant sert à ouvrir une connexion à une autre base de données MySQL, *project_q2*.

3. Double-cliquez sur le **tMysqlInput** afin d'ouvrir sa vue **Basic settings**.

4. Cochez la case **Use an existing connection** et laissez la case **Component List** comme elle est.

5. Cliquez sur le bouton **[...]** situé à côté du champ **Edit schema** afin d'ouvrir la boîte de dialogue **[Schema]**. Définissez la structure de la table de la base de données à partir de laquelle les données sont lues.

Dans cet exemple, la structure de la table de la base de données est composée de quatre colonnes : *id* (de type **Integer** et d'une longueur de 2 caractères), *firstName* (de type **String** et d'une longueur de 15 caractères), *lastName* (de type **String** et d'une longueur de 15 caractères) et *city* (de type **String** et d'une longueur de 15 caractères). Cliquez sur **OK** pour fermer la boîte de dialogue et propager le schéma au composant suivant.

6. Dans le champ **Table**, saisissez le nom de la table de la base de données, *customers* dans cet exemple. Cliquez sur **Guess Query** afin de générer l'instruction de la requête correspondant au schéma de votre table dans le champ **Query**.

7. Dans la vue **Dynamic settings**, cliquez sur le bouton **[+]** afin d'ajouter une ligne à votre table. Dans le champ **Code**, saisissez le code du script de la variable de contexte que vous venez de créer, `" + context.myConnection + "` dans cet exemple.

8. Dans la vue **Basic settings** du **tLogRow**, sélectionnez l'option **Table** pour un meilleur affichage des résultats d'exécution du Job.

9. Dans la vue **Dynamic settings** du **tMysqlClose**, procédez de la même manière que dans la vue **Dynamic settings** du **tMysqlInput**.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job. Appuyez sur **F6** ou cliquez sur **Run** afin de l'exécuter.

   Une boîte de dialogue s'affiche vous demander de spécifier le composant de connexion que vous souhaitez utiliser.

2. Sélectionnez le composant de connexion **tMysqlConnection_1** et cliquez sur **OK**.

   Les données lues depuis la base de données *project_q1* s'affichent dans la console **Run**.

3. Appuyez sur **F6** ou cliquez sur **Run** afin d'exécuter le Job à nouveau. Quand la boîte de dialogue s'affiche, sélectionnez l'autre composant de connexion, à savoir **tMysqlConnection_2**, afin de lire les données depuis l'autre base de données, *project_q2*.

   Les données lues depuis la base de données *project_q2* s'affichent dans la console **Run**.

tMysqlLastInsertId

Propriétés du tMysqlLastInsertId

Famille de composant	Databases/MySQL	
Fonction	**tMysqlLastInsertId** affiche les derniers ID ajoutés à une table à partir d'une connexion MySQL spécifiée.	
Objectif	**tMysqlLastInsertId** permet de récupérer les clés primaires des entrées dernièrement ajoutées par un utilisateur à une table MySQL.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Component list	Sélectionnez le composant **tMysqlConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant intermédiaire. *Si vous utilisez ce composant avec un **tMysqlOutput**, vérifiez que la case **Extend Insert** de l'onglet **Advanced settings** est bien décochée. En effet, **Extend Insert** permet de faire un chargement par lot, ainsi seul l'ID de la dernière ligne du dernier lot sera retourné.*
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Récupérer les ID des dernières entrées ajoutées

Le scénario Java suivant permet d'ouvrir une connexion à une base de données MySQL, d'y écrire des données spécifiées et enfin de récupérer les derniers ID ajoutés à la connexion existante.

- Cliquez sur chacun des composants suivants et déposez-les dans l'éditeur graphique : **tMysqlConnection**, **tMysqlCommit**, **tFileInputDelimited**, **tMysqlOutput**, **tMysqlLastInsertId**, et **tLogRow**.
- Reliez le **tMysqlConnection** au **tFileInputDelimited** à l'aide d'un lien **OnSubjobOk**.
- Reliez le **tFileInputDelimited** au **tMysqlCommit** à l'aide d'un lien **OnSubjobOk**.
- Reliez les trois derniers composants au **tFileInputDelimited** et entre eux à l'aide de liens **Row Main**, comme suit :

- Sélectionnez le **tMysqlConnection** en cliquant dessus à partir de l'éditeur graphique.
- Cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**).
- Dans l'onglet **Basic settings**, entrez les paramètres de connexion manuellement ou sélectionnez-les à partir d'une liste de variables de contexte, si vous les avez stockées localement sous le nœud **DB connections** de la partie **Metadata** (Métadonnées) du **Repository**. Vous accéderez à cette liste en cliquant dans le champ à renseigner puis en appuyant sur **Ctrl+Espace**. Pour plus d'informations sur les Métadonnées, consultez le *Guide utilisateur du Studio Talend*.

- Sélectionnez le **tMysqlCommit** en cliquant dessus à partir de l'éditeur graphique, puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**).
- Dans le champ **Component list**, sélectionnez le composant **tMysqlConnection** approprié si vous effectuez un Job à plusieurs connexions.
- Sélectionnez le **tFileInputDelimited** en cliquant dessus à partir de l'éditeur graphique.
- Cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**).

- Renseignez le chemin d'accès au fichier à traiter dans le champ **File Name**. Dans cet exemple, le nom du fichier est *Customers*.
- Dans le champ **Row separator**, définissez le séparateur de lignes permettant d'identifier les fins de ligne. De même, pour **Field separator**, définissez le séparateur de champs permettant de délimiter les différents champs pour chaque ligne.
- Définissez, si nécessaire, le nombre de lignes d'en-tête et de pied-de-page à ignorer, ainsi que le nombre maximum de lignes que vous souhaitez traiter. Dans cet exemple, le fichier comporte une ligne d'en-tête.

- Cliquez sur le bouton **[...]** à côté de la mention **Edit Schema** pour définir les données à passer au composant suivant.

Sujets associés : consultez le *Guide utilisateur du Studio Talend*.

Dans ce scénario, le schéma se compose de deux colonnes, *name* et *age*. La première colonne comprend le nom des employés et la seconde comporte leurs âges respectifs.

- Sélectionnez le **tMysqlOutput** en cliquant dessus à partir de l'éditeur graphique.

- Cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**).

- Cochez la case **Use an existing connection**.

- Dans le champ **Table**, renseignez le nom de la table dans laquelle vous voulez écrire la liste des employés. Dans cet exemple, il s'agit de la table *employee*.

- Sélectionnez les actions à effectuer dans les champs **Action on table** et **Action on data**, à partir des listes proposées. Dans cet exemple, n'effectuez aucune action sur la table elle-même mais exécutez une action *Insert* sur les données.

- Cliquez sur le bouton **Sync columns** pour synchroniser les colonnes par rapport au composant précédent. Dans cet exemple, le schéma à reproduire dans la table de la base de données MySQL se compose des deux colonnes *name* et *age*.

- Sélectionnez le **tMysqlLastInsertId** en cliquant dessus à partir de l'éditeur graphique.
- Cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**).

- Dans le champ **Component list**, sélectionnez le composant **tMysqlConnection** approprié si vous effectuez un Job à plusieurs connexions.
- Cliquez sur le bouton **Sync columns** pour synchroniser les colonnes par rapport au composant précédent. Dans le schéma de sortie du **tMysqlLastInsertId**, vous pouvez voir la colonne last_insert_id qui indique les dernières ID ajoutées à la connexion existante.

Vous pouvez sélectionner le type de données *Long* dans la liste déroulante **Type**, dans le cas de nombreuses entrées.

- Sélectionnez le **tLogRow** en cliquant dessus à partir de l'éditeur graphique, puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**). Pour plus d'informations, consultez le *tLogRow*.

- Sauvegardez votre Job puis appuyez sur **F6** pour l'exécuter.

```
Starting job lastinserted at 11:12 01/09/2008.
.--------+---+--------------.
|         tLogRow_1          |
|=-------+---+-------------=|
|name    |age|last_insert_id|
|=-------+---+-------------=|
|Marie   |24 |40            |
|Patrick |22 |41            |
|Pierrick|27 |42            |
'--------+---+--------------'

Job lastinserted ended at 11:12 01/09/2008. [exit code=0]
```

Le composant **tMysqlLastInsertId** affiche les dernières ID ajoutées pour chaque ligne de la connexion existante.

tMysqlOutput

Propriétés du tMysqlOutput

Famille de composant	Databases/MySQL	
Fonction	Le composant **tMysqlOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tMysqlOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez la version de MySQL que vous utilisez.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **Default** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore.

		Create table if doesn't exist : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. **Insert or update on duplicate key or unique index** : Ajouter des entrées si la valeur insérée n'existe pas ou mettre à jour les entrées si la valeur insérée existe déjà et qu'un risque de violation d'une clé unique se présente. **Insert ignore** : Ajouter uniquement de nouvelles lignes afin d'empêcher les erreurs de doublons de clés. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et cochez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si

		vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. *Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l'alias de la source de données spécifié.* Cette option est indisponible lorsque la case **Use an existing connection** est cochée.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Extend Insert	Cochez cette case pour insérer un ensemble de lignes définies au lieu d'insérer les lignes une par une. Cette option permet un important gain de performance. **Number of rows per insert** : saisissez le nombre de lignes à insérer en un bloc. Notez que si vous sélectionnez un nombre important de lignes, cela peut augmenter la quantité de mémoire utilisée et donc diminuer les performances. Cette option n'est pas compatible avec le lien **Reject**. Vous devez donc la décocher si vous utilisez un lien **Reject** en sortie du composant.
	Use Batch size	Cochez cette case afin d'activer le mode batch pour le traitement des données. Dans le champ **Batch Size** qui apparaît, vous pouvez saisir le nombre de lignes par lot à charger. Cette case est disponible uniquement si vous avez choisi l'option **Insert**, **Update**, **Single Insert Query** ou **Delete** dans le champ **Action on data**.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tMysqlOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Use Hint Options	Cochez cette case pour activer la zone de configuration des indicateurs (ou Hints) permettant d'optimiser l'exécution d'une requête. Dans cette zone, les paramètres sont : - **HINT** : spécifiez l'indicateur dont vous avez besoin, en utilisant la syntaxe `/*+ */`. - **POSITION** : spécifiez la place de l'indicateur dans une instruction SQL. - **SQL STMT** : sélectionnez l'instruction SQL que vous souhaitez utiliser.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	Use duplicate key update mode insert	Met à jour les valeurs des colonnes spécifiées en cas de doublon de la clé primaire. **Column** : Saisissez entre guillemets le nom de la colonne à mettre à jour.

		Value : Saisissez l'opération que vous souhaitez effectuer sur la colonne. Pour utiliser cette option, vous devez préalablement sélectionner le mode **Insert** dans la liste **Action on data** de la vue **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Ce composant doit être utilisé en tant que composant de sortie. Il permet de faire des actions sur une table ou les données d'une table d'une base de données MySQL. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation du **tMysqlOutput**, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Ajouter une colonne et modifier les données

Ce scénario Java est un Job à trois composants permettant de créer de manière aléatoire des données de test à l'aide du composant **tRowGenerator**, dupliquer la colonne à modifier à l'aide du composant **tMap** et de modifier les données à insérer en utilisant une expression SQL dans le composant **tMysqlOutput**.

- Cliquez et déposez les composants suivants dans l'éditeur graphique : **tRowGenerator**, **tMap** et **tMysqlOutput**.
- Connectez-les à l'aide de liens **Row Main**.

- Dans l'éditeur graphique, double-cliquez sur le **tRowGenerator** pour afficher l'onglet **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir les données à transmettre au composant **tMap**, dans ce scénario, les deux colonnes *name* et *random_date*.

- Cliquez sur **OK** pour fermer la boîte de dialogue.

- Cliquez sur le bouton **[...]** à côté du champ **RowGenerator Editor** pour ouvrir l'éditeur du **tRowGenerator** pour paramétrer les données à générer.

- Cliquez dans le champ **Functions** correspondant et sélectionnez une fonction pour chacune des deux colonnes, `getFirstName` pour la colonne *name* et `getrandomDate` pour la colonne *random_date*.
- Dans le champ **Number of Rows for Rowgenerator**, saisissez 10 pour ne générer que dix lignes de prénoms et cliquez sur **Ok** pour fermer l'éditeur.
- Double-cliquez sur le composant **tMap** pour ouvrir l'éditeur du **tMap**. L'éditeur s'ouvre et affiche les données d'entrée du composant **tRowGenerator**.

- Dans le panneau **Schema editor** de l'éditeur du **tMap**, cliquez sur le bouton **[+]** de la table de sortie pour ajouter deux lignes et nommez la première colonne *random_date* et la deuxième *random_date1*.

Dans ce scénario, l'objectif est de dupliquer la colonne *random_date* et adapter le schéma afin de modifier les données à transférer dans le composant de sortie.

- Dans l'éditeur du **tMap**, glissez la ligne *random_date* de la table d'entrée vers les lignes *random_date* et *random_date1* de la table de sortie.

- Cliquez sur **OK** pour fermer l'éditeur.
- Dans l'éditeur graphique, double-cliquez sur le composant **tMysqlOutput** pour afficher l'onglet **Basic settings** et paramétrer ses propriétés.

- Dans la liste **Property Type**, sélectionnez l'option **Repository** et cliquez sur le bouton **[...]** pour ouvrir la boîte de dialogue **[Repository content]** et sélectionnez la connexion à la base de données adéquate. Les informations de connexion sont renseignées automatiquement.
- Configurez les informations de connexion à la base de données dans les champs correspondants.

Si vous n'avez pas stocké de connexion à votre base de données sous le nœud **DB connections** sous le nœud **Metadata** du **Repository**, sélectionnez l'option **Built-in** dans la liste **Property Type** et renseignez les informations de connexion manuellement.

- Cliquez sur le bouton **[...]** à côté du champ **Table** et sélectionnez la table à modifier : la table *Dates* dans ce scénario.

- Dans la liste **Action on table**, sélectionnez l'option **Drop table if exists and create** et sélectionnez *Insert* dans la liste **Action on data**.
- Si nécessaire, cliquez sur **Sync columns** pour récupérer le schéma de colonnes du composant **tMap**.
- Cliquez sur l'onglet **Advanced settings** pour afficher la vue correspondante et définir les paramètres avancés du composant.

- Dans la zone **Additional Columns**, paramétrez les modifications à appliquer aux colonnes.

  Dans ce scénario, la colonne *One_month_later* remplace la colonne *random_date_1* et les données aussi sont modifiées à l'aide d'une requête SQL ajoutant un mois à la date générée aléatoirement dans la colonne *random_date_1*. Par exemple : 2007-08-12 devient 2007-09-12.

  -Saisissez *One_Month_Later* dans la cellule **Name**.

  -Dans la cellule **SQL expression**, saisissez la requête SQL ajoutant un mois, dans ce scénario : `"adddate(Random_date, interval 1 month)"`.

  -Sélectionnez **Replace** dans la liste **Position**.

  -Sélectionnez *Random_date1* dans la liste **Reference column**.

Pour ce Job, vous dupliquez la colonne *random_date_1* dans la table avant de la remplacer par la colonne *One_Month_Later*. Le but de cette manipulation était de voir les modifications apportées en amont.

- Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

La nouvelle colonne *One_month_later* remplace la colonne *random_date1* dans la base de données et ajoute un mois à chaque date générée aléatoirement.

Random_date	One_Month_Later
2007-03-13 21:52:09	2007-04-13 21:52:09
2008-11-02 15:33:29	2008-12-02 15:33:29
2008-05-17 18:47:15	2008-06-17 18:47:15
2007-09-06 17:44:36	2007-10-06 17:44:36
2008-11-17 11:46:36	2008-12-17 11:46:36
2007-05-19 00:46:58	2007-06-19 00:46:58
2008-04-20 15:36:00	2008-05-20 15:36:00
2007-07-24 21:06:08	2007-08-24 21:06:08
2008-04-03 13:24:17	2008-05-03 13:24:17
2007-06-29 13:22:23	2007-07-29 13:22:23

Voir également *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.

Scénario 2 : Effectuer des mises à jour dans une base de données

Ce scénario Java est un Job à deux composants permettant de mettre à jour les données d'une table MySQL à partir d'un fichier délimité.

- A partir de la **Palette**, cliquez et déposez les composants **tFileInputDelimited** et **tMysqlOutput** dans l'éditeur graphique puis connectez-les à l'aide d'un lien **Row Main**.

- Dans l'éditeur graphique, double-cliquez sur le **tFileInputDelimited** pour afficher l'onglet **Basic settings**.

- Dans le champ **Property Type**, sélectionnez le mode **Repository** si vous avez stocké les informations du fichier délimité sous le nœud **Metadata** du **Repository** ou sélectionnez le mode **Built-In** pour les définir manuellement. Dans ce scénario, utilisez le mode **Built-In**.

- Dans le champ **File Name**, renseignez manuellement le chemin d'accès au fichier contenant les mises à jour à propager dans la base de données ou cliquez sur le bouton **[...]** pour parcourir vos dossiers jusqu'à ce fichier. Dans cet exemple, utilisez le fichier *customer_update*, qui comporte quatre colonnes, *id*, *CustomerName*, *CustomerAddress* et *idState*.

id	CustomerName	CustomerAddress	idState
858	Froggy's Gourmet Catering	1831 Beverly Place #9D	4
859	Dependable Plumbing and Sewer	1550 Ridge Rd.	25
860	Lickmen Restoration	1235 Easton Rd.	40
861	Acturial Enterprises Ltd.	3148 cottonwood Ct.	18
862	Rythmics Ltd.	857 Woodbine Rd	30
863	Acturial Enterprises Ltd.	1482 Concorde Circle	48
864	Crosstracks Car Wash	218 Oakridge Ave.	39
865	Meonits & Mogogni Inc.	616 Cobblestone Cir.	17
866	Foy Aviation	2220 Grant Blvd.	50
867	Ebert Music Center	12 Broadview Lane	29
868	janice Mann Accounting Service	1660 Park Ave.	9
869	Johnson, Erico & Co CPA's	2922 Twin Oaks Drive	40
870	Corbins, Rodriguez, & Savocchi	115 Pleasent Ave.	18
871	Nina's Snow Plowing	3385 University Ave.	20
872	Darcy Frame and Matting Servic	1101 Deerfield Place	47
873	Marks, Marks, and Kaplan Ltd.	1949 Cloverdale Rd.	9

- Définissez si nécessaire les séparateurs de lignes (**Row separator**) et de champs (**Field separator**), l'en-tête (**Header**) et le pied-de-page (**Footer**), ainsi que le nombre de lignes à traiter (**Limit**). La première ligne, qui porte le nom des colonnes, est ignorée dans cet exemple, c'est pourquoi le champ **Header** est défini à "1".

- Cliquez sur le bouton **[...]** à côté du champ **Edit Schema** pour définir les données à passer au composant suivant. Dans cet exemple, le schéma est constitué de quatre colonnes, *id*, *CustomerName*, *CustomerAddress* et *idState*.

Column	Key	Type	Nullable
id	☑	Integer	☑
CustomerName	☐	String	☑
CustomerAddress	☐	String	☑
idState	☐	Integer	☑

- Devant le nom de chaque colonne se trouve une case *Key*. Cochez cette case pour la ou les colonne(s) que vous voulez définir comme clé.

Vous devez obligatoirement définir au moins une clé pour que le Job s'exécute. Dans le cas contraire, le Job s'arrête automatiquement et un message d'erreur s'affiche dans la console de log.

- Dans l'éditeur graphique, double-cliquez sur le composant **tMysqlOutput** pour paramétrer ses propriétés dans l'onglet **Basic settings** de la vue **Component** :

- Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton **[...]** à côté du champ **Edit schema** pour consulter le schéma et vérifier qu'au moins une colonne a été définie comme clé.

- Dans le champ **Property Type**, sélectionnez **Repository** si vous avez stocké les informations de connexion à la base de données sous le nœud **Metadata** du **Repository** ou sélectionnez le mode **Built-In**, puis renseignez manuellement les champs suivants : **Host**, **Port**, **Database**, **Username** et **Password**.

- Dans le champ **Table**, saisissez le nom de la table à mettre à jour.

- Dans le champ **Action on table**, sélectionnez l'opération que vous souhaitez effectuer sur la table. Pour ce scénario, sélectionnez **Default** car la table existe déjà.

- Dans le champ **Action on data**, sélectionnez l'opération que vous souhaitez effectuer sur les données. Pour ce scénario, sélectionnez l'option **Update** pour mettre la table à jour.

- Enregistrez le Job puis appuyez sur **F6** pour l'exécuter.

id	CustomerName	CustomerAddress	idState
858	Froggy's Gourmet Catering	1831 Beverly Place #9D	4
859	Dependable Plumbing and Sewer	1550 Ridge Rd.	25
860	Lickmen Restoration		
861	Acturial Enterprises Ltd.		
862	Rythmics Ltd.		
863	Acturial Enterprises Ltd.		
864	Crosstracks Car Wash		
865	Meonits & Mogogni Inc.		
866	Foy Aviation		
867	Ebert Music Center		
868	janice Mann Accounting Servic		
869	Johnson, Erico & Co CPA's		
870	Corbins, Rodriguez, & Savocch		
871	Nina's Snow Plowing		
872	Darcy Frame and Matting Serv		
873	Marks, Marks, and Kaplan Ltd.		

id	CustomerName	CustomerAddress	idState
858	Froggy's Gourmet Catering	1831 Beverly Place #9-11D	4
859	Dependable Plumbing and Sewer	1550 Ridge Rd.	25
860	Lickmen Restoration	1235 Easton Rd.	40
861	Acturial Enterprises Ltd.	3148 Cottonwood Ct.	18
862	Rythmics Ltd.	857 Woodbine Rd	30
863	Acturial Enterprises Ltd.	1482 Concorde Circle	48
864	Crosstracks Car Wash	218 Oakridge Ave.	39
865	Meonits & Mogogni Inc.	616 Cobblestone Cir.	17
866	Foy Aviation	2220 Grant Blvd.	50
867	Ebert Music Center	12 Broadview Lane	29
868	janice Mann Accounting Service	1660 Park Ave.	9
869	Johnson, Erico & Co CPA's	2922 Twin Oaks Drive	40
870	Corbins, Rodriguez, & Savocchi	115 Pleasent Ave.	18
871	Nina's Snow Plowing	3385 University Ave.	20
872	Darcy Frame and Matting Servic	1101 Deerfield Place	47
873	Marks, Kaplan and Jones Ltd.	1949 Cloverdale Rd.	9

A partir de votre explorateur de base de données, vous pouvez vérifier que la table *customers* a bien été mise à jour. Comme avant sa mise à jour, elle se présente sous la forme des quatre colonnes *id*, *CustomerName*, *CustomerAddress* et *idState*, mais certains champs ont été modifiés selon les données du fichier délimité *customer_update*.

Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject

Ce scénario décrit un Job à quatre composants effectuant une migration d'un fichier client vers la table d'une base de données MySQL et redirigeant les données erronées vers un fichier CSV à l'aide d'un lien **Reject**.

- Dans le **Repository**, sélectionnez la métadonnée correspondant au fichier client que vous souhaitez migrer et glissez-la dans l'espace de modélisation. Dans la boîte de dialogue **[Components]**, sélectionnez **tFileInputDelimited** et cliquez sur **OK**. Les propriétés du composant seront automatiquement renseignées.

- Si vous n'avez pas enregistré les informations concernant votre fichier client sous le nœud **Metadata** du **Repository**. Glissez un composant **tFileInputDelimited** de la famille **File** > **Input** de la **Palette** et renseignez ses propriétés manuellement dans la vue **Component**.

- A partir de la **Palette**, glissez un composant **tMap** de la famille **Processing** dans l'espace de modélisation.

- Dans le **Repository**, développez le nœud **Metadata** et **Db Connections,** et sélectionnez la métadonnée de connexion à la base de données dans laquelle vous souhaitez migrer vos données et glissez-la dans l'espace de modélisation. Dans la boîte de dialogue **[Components]**, sélectionnez **tMysqlOutput** et cliquez sur **OK**. Les propriétés de connexion à la base de données seront automatiquement renseignées.

- Si vous n'avez pas enregistré les informations de connexion à votre base de données sous le nœud **Db Connections** du **Repository**, glissez un composant **tMysqlOutput** de la famille **Databases** de la **Palette** et renseignez ses propriétés manuellement dans la vue **Component**.

Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.

- A partir de la **Palette**, sélectionnez un composant **tFileOutputDelimited** de la famille **File** > **Output** et glissez-le dans l'espace de modélisation.

- Reliez les composant **customers** et **tMap**, et les composants **tMap** et **Localhost** à l'aide d'un lien de type **Row Main**. Vous nommerez ce deuxième lien *out*.

- Reliez le composant **Localhost** au **tFileOutputDelimited** à l'aide d'un lien de type **Row** > **Reject**.

- Double-cliquez sur le composant **customers** pour afficher la vue **Component** correspondante.

- Dans la liste **Property Type**, sélectionnez le mode **Repository** puis cliquez sur le bouton **[...]** correspondant pour sélectionner la métadonnée contenant la connexion à votre fichier. Sinon, sélectionnez le mode **Built-in** et paramétrez manuellement les champs suivants.

- Cliquez sur le bouton **[...]** à côté du champ **File Name**, et indiquez le chemin d'accès et le nom du fichier à utiliser.

- Dans les champs **Row** et **Field Separator**, saisissez entre guillemets les séparateurs de lignes et de champs utilisés dans le fichier.

- Dans les champs **Header**, **Footer** et **Limit**, saisissez le nombre de ligne d'en-tête et de pied-de-page à ignorer, ainsi que le nombre de ligne limite à traiter.

- Dans la liste **Schema**, sélectionnez **Repository** et cliquez sur le bouton **[...]** correspondant pour sélectionner le schéma de votre fichier, s'il est stocké sous le nœud **Metadata** du **Repository**. Vous pouvez aussi cliquer sur le bouton **[...]** correspondant au champ **Edit schema** et définir manuellement le schéma de votre fichier.

Le schéma se présente comme suit :

- Double-cliquez sur le **tMap** afin d'ouvrir son éditeur.

- Sélectionnez les colonnes *id*, *CustomerName*, *CustomerAddress*, *idSate*, *id2*, *RegTime*, *RegisterTime* de la zone de gauche et glissez-les dans la table **out** de la zone de droite.

- Dans la zone **Schema editor** en bas de l'éditeur du **tMap**, dans le tableau à droite, changez la longueur de la colonne **CustomerName** en *28* afin de créer une erreur. Ainsi, les données dont la longueur est supérieure à 28 créeront des erreurs qui pourront être récupérées grâce au lien **Reject**.
- Cliquez sur **OK**.
- Dans l'espace de modélisation, double-cliquez sur le composant de sortie **Localhost** pour afficher la vue **Component** correspondante.

- Dans la liste **Property Type**, sélectionnez le mode **Repository** puis cliquez sur le bouton **[...]** correspondant pour sélectionner la métadonnée de connexion à la base de données. Les informations de connexion sont renseignées automatiquement. Sinon, sélectionnez le mode **Built-in** et paramétrez ces champs manuellement.
- Dans le champ **Table**, saisissez le nom de la table à créer. Dans ce scénario, vous l'appelez *customers_data*.
- Dans la liste **Action on data**, sélectionnez l'option **Create table**.
- Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
- Assurez-vous que la case **Die on error** est bien décochée, afin que le Job s'exécute malgré l'erreur que vous venez de créer.

- Cliquez sur l'onglet **Advanced settings** de la vue **Component** pour paramétrer les paramètres avancés du composant.

- Décochez la case **Extend Insert** permettant d'insérer des lignes par bloc car cette option n'est pas compatible avec le lien **Reject**.

- Double-cliquez sur le composant **tFileOutputDelimited** pour paramétrer ses propriétés dans la vue **Component**.

- Cliquez sur le bouton **[...]** à côté du champ **File Name** pour indiquer le répertoire de destination et le nom du fichier de sortie.

- Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

1	Griffith Paving	talend@apres	7	41	###########	2001-01-17 06:28	Data truncation: Data too long
5	Terrinni & Sor	770 Exmoor F	5	9	###########	1982-04-19 10:28	Data truncation: Data too long
10	Elle Hypnosis	2032 Northbro	1	7	###########	1975-06-10 20:20	Data truncation: Data too long
15	Darcy Frame	1633 McGove	21	28	###########	2001-01-07 20:40	Data truncation: Data too long
19	Salt & Peppe	965 Marion P	44	33	###########	2006-01-24 10:53	Data truncation: Data too long
31	Arthur M. Fei	1220 Heather	37	24	###########	1974-07-24 20:48	Data truncation: Data too long
36	Got It All Elec	629 Kincaid A	22	28	###########	1993-10-22 15:28	Data truncation: Data too long
53	Crossroads C	2267 Andersc	5	14	###########	1977-10-24 17:53	Data truncation: Data too long
54	Jay's Cabinet	2024 St.John	48	48	2006-02-29 06	2001-01-10 07:48	Data truncation: Data too long
57	Lee Stairwork	2644 Princeto	41	46	###########	1992-04-15 09:41	Data truncation: Data too long
58	Lambert & Lo	662 Lyons Ci	45	17	###########	2002-11-22 04:28	Data truncation: Data too long
70	janice Mann A	3181 Barkwo	28	35	###########	1983-04-20 15:20	Data truncation: Data too long
72	Michael Mont	1077 Court A	9	9	###########	2000-10-07 00:20	Data truncation: Data too long
83	Dean Barnett	2922 Twin Oa	18	46	###########	2006-03-23 10:14	Data truncation: Data too long
96	Doerner's Aw	616 Cobblest	26	28	###########	1993-05-03 04:33	Data truncation: Data too long
##	Auto Interior (	844 Spruce S	48	40	###########	2006-04-25 13:14	Data truncation: Data too long
##	First National	465 Stratford	5	41	###########	2003-09-23 01:54	Data truncation: Data too long
##	Terrinni & Sor	1859 Green E	12	44	###########	1986-12-26 05:48	Data truncation: Data too long

Les données en erreur sont envoyées dans le fichier délimité, ainsi que le type d'erreur rencontrée. Ici, vous avez affaire à des données tronquées : **Data truncation**.

tMysqlOutputBulk

Propriétés du tMysqlOutputBulk

Les composants **tMysqlOutputBulk** et **tMysqlBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tMysqlOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/MySQL	
Fonction	Ecrit un fichier composé de colonnes et basé sur le délimiteur défini et sur les standards MySQL.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données MySQL.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer. Ce fichier est généré sur la même machine que le *Studio Talend* ou que le Job contenant un **tMysqlOutputBulk**.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager

		ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Row separator*	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Text enclosure	Caractères utilisés pour entourer le texte.
	Create directory if does not exist	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Custom the flush buffer size	Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ **Row number**, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée.
	Records contain NULL value	Cette case est cochée par défaut. Elle permet de prendre les champs de valeur NULL en compte. Si vous la décochez, les valeurs NULL seront remplacées par des valeurs vides.
	Check disk space	Cochez cette case afin de retourner une exception durant l'exécution si le disque est plein.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tMysqlBulkExec**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données MySQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Insérer des données transformées dans une base MySQL

Ce scénario décrit un Job contenant quatre composants dont le but est d'alimenter un base MySQL à partir d'un fichier paramètres contenant des données transformées. Deux étapes sont requises pour ce Job, d'une part pour créer

le fichier paramètres qui sera utilisé dans un second temps. La première étape inclut une phase de transformation des données contenues dans le fichier.

Déposer et relier les composants

1. Glissez et déposez les composants suivants : **tRowGenerator**, **tMap**, **tMysqlOutputBulk** et **tMysqlBulkExec** dans l'espace de modélisation.

2. Connectez le flux principal à l'aide de connexions **Row > Main**.

3. Connectez le composant de début (**tRowGenerator**, dans cet exemple) au composant **tMysqlBulkExec** à l'aide d'une connexion **Trigger** de type **OnSubjobOk**.

Configurer les composants

1. Un composant **tRowGenerator** est utilisé pour générer des données de façon aléatoire. Double-cliquez sur le composant **tRowGenerator** pour lancer l'éditeur.

2. Définissez le schéma des lignes à générer et la nature des données à générer. Dans cet exemple, le fichier *clients* à créer contient les colonnes suivantes : *ID*, *First Name*, *Last name*, *Address*, *City* qui sont toutes de type chaîne de caractères (**string**) à l'exception de l'*ID* qui est de type entier (**integer**).

Certaines informations du schéma n'ont pas nécessairement besoin d'être affichées. Pour les dissimuler, cliquez sur le bouton **Columns** dans la barre d'outils et décochez les colonnes à cacher, par exemple : **Precision** ou **Parameters**.

Utilisez le bouton **[+]** pour ajouter autant de colonnes que possible à votre schéma.

Cliquez sur le bouton **Refresh** en haut à droite de l'éditeur pour visualiser un aperçu des lignes générées en sortie dans l'onglet **Preview** en bas de l'éditeur.

3. Dans l'espace de modélisation, double-cliquez sur le composant **tMap** pour ouvrir son éditeur et paramétrer la transformation.

4. Faites glisser toutes les colonnes de la table d'entrée (**row1**) vers la table de sortie (**clients**).

5. Appliquez la transformation sur la colonne *LastName* en ajoutant `.toUpperCase()` à la fin. Cela mettra les noms des clients en majuscule.

   Cliquez sur **OK** pour valider la transformation.

6. Dans l'espace de modélisation, double-cliquez sur le composant **tMysqlOutputBulk** pour afficher sa vue **Component** et paramétrer ses propriétés.

7. Définissez le chemin d'accès et le nom du fichier à produire dans le champ **File Name**. Si les propriétés du fichier délimité sont conservées dans le **Repository**, sélectionnez l'entrée correspondante dans le champ **Property type** afin de les récupérer. Dans cet exemple, le nom de fichier est *clients.txt*.

Le schéma est propagé à partir du composant **tMap**, si vous cliquez sur **Yes** lorsque la boîte de dialogue vous demande de confirmer la propagation. Sinon, cliquez sur le bouton **Sync columns** pour récupérer le schéma.

8. Dans cet exemple, n'incluez pas les informations d'en-tête, puisque la table doit déjà les contenir.

9. Cliquez sur **OK** pour valider la sortie.

10. Double-cliquez sur le composant **tMysqlBulkExec** pour paramétrer la requête INSERT à exécuter.

11. Paramétrez les informations de connexion. Il est recommandé de stocker ce type d'informations dans le **Repository**, afin de pouvoir les récupérer à tout moment pour tout Job.

12. Dans le champ **Table**, saisissez le nom de la table à alimenter, ici, *clients*.

13. Dans l'onglet **Advanced settings**, saisissez le séparateur de colonnes, dans le champ **Fields terminated by**.

14. Vérifiez que l'encodage défini dans le champ **Encoding** correspond à celui de vos données.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

Resultset 1

ID	Firstname	Lastname	Address	City
1	Theodore	EISENHOWER	Moreno Drive	Concord
2	Rutherford	COOLIDGE	Lindbergh Blvd	Madison
3	William	JEFFERSON	Katella Avenue	Boise
4	Benjamin	BUCHANAN	French Camp Turnpike Road	Juneau
5	Andrew	WILSON	North Preisker Lane	Frankfort
6	Theodore	MCKINLEY	Pacific Hwy S	Augusta
7	William	POLK	Calle Real	Tallahassee
8	Dwight	HAYES	Timberlane Drive	Des Moines
9	Abraham	BUCHANAN	Padre Boulevard	Des Moines
10	Abraham	MADISON	N Kentwood	Olympia

La table de la base *clients* est alimentée avec les données du fichier notamment les données transformées dans la colonne *Last name*.

Pour une simple opération d'Insert ne nécessitant pas de transformation, l'utilisation du composant **tMysqlOutputBulkExec** permet d'économiser une étape dans le processus et ainsi de gagner en performance.

Voir également *Propriétés du tMysqlOutputBulkExec*.

tMysqlOutputBulkExec

Propriétés du tMysqlOutputBulkExec

Les composants **tMysqlOutputBulk** et **tMysqlBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tMysqlOutputBulkExec**.

Famille de composant	Databases/MySQL	
Fonction	Le **tMysqlOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tMysqlOutputBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données MySQL.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez la version de MySQL que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectue aucune action sur la table. **Drop and create table** : supprime la table puis en créer une nouvelle. **Create table** : crée une table qui n'existe pas encore. **Create table if not exists** : crée la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprime le contenu de la table.
	Table	Nom de la table à écrire. Une seule table peut être écrite à la fois et cette table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Local FileName	Nom du fichier à générer et à traiter. Ce fichier est généré sur la même machine que le *Studio Talend* ou que le Job contenant un **tMysqlOutputBulkExec**, puis chargé dans la base de données spécifiée dans le champ **Host**.

	Append	Cochez cette option pour ajouter de nouvelles lignes à la fin du fichier.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
Advanced settings	*Additional JDBC Parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Text enclosure	Caractères entourant les données.
	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Custom the flush buffer size	Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ **Row number**, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert records in table** : Insère de nouveaux enregistrements dans une table. **Replace records in table** : Remplace les enregistrements existants par de nouveaux. **Update records in table** : Modifie les enregistrements existants. **Ignore records in table** : Ignore les enregistrements existants ou insère les nouveaux.

	Records contain NULL value	Cette case est cochée par défaut. Elle permet de prendre les champs de valeur NULL en compte. Si vous la décochez, les valeurs NULL seront remplacées par des valeurs vides.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	

Scénario : Insérer des données dans une base MySQL

Ce scénario est un Job à deux composants qui effectue le même type d'opération d'alimentation que les composants **tMysqlOutputBulk** (consultez *tMysqlOutputBulk*) et **tMysqlBulkExec** (consultez *tMysqlBulkExec*), mais sans transformation de données.

- Cliquez et déposez les composants suivants : **tRowGenerator** et **tMysqlOutputBulkExec**.
- Connectez les composants via un lien de type **Row** > **Main**.
- Paramétrez le **tRowGenerator** de la même manière que dans *Scénario : Insérer des données transformées dans une base MySQL*. Le schéma est constitué de plusieurs colonnes notamment : *ID*, *First Name*, *Last Name*, *Address* et *City*.
- A partir de l'espace de modélisation, double-cliquez sur le composant **tMysqlOutputBulkExec** pour afficher la vue **Component** et paramétrer ses propriétés.

- Paramétrez la connexion à la base de données si nécessaire, en suivant les recommandations dans *Scénario : Insérer des données transformées dans une base MySQL*, concernant la conservation des informations de connexion dans la partie **Metadata** du **Repository**. Pour cela, sélectionnez **Repository** dans le champ **Property Type** et sélectionnez la connexion adéquate dans le champ adjacent. Les champs suivants seront renseignés automatiquement.

Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.

- Dans le champ **Action on table**, sélectionnez l'option **None** puisque vous souhaitez insérer les données dans une table déjà existante.
- Dans le champ **Table**, saisissez le nom de la table à alimenter, ici, *clients*.
- Dans le champ **Local filename**, indiquez le chemin d'accès et le nom du fichier contenant les données à charger dans la table, ici, *clients.txt*.
- Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
- Cliquez sur l'onglet **Advanced settings** pour paramétrer les propriétés avancées du composant.

- Dans la liste **Action on data**, sélectionnez l'option **Insert records in table** pour insérer ces nouvelles données dans la table.
- Appuyez sur **F6** pour exécuter le Job

Le résultat devrait être relativement identique à celui obtenu dans *Scénario : Insérer des données transformées dans une base MySQL*. Cependant les données elles-mêmes peuvent différer légèrement puisque les données sont regénérées de façon aléatoire à chaque exécution.

tMysqlRollback

Propriétés du tMysqlRollback

Famille de composant	Databases/MySQL	
Fonction	Le **tMysqlRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tMysqlConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants MySQL, notamment les composants **tMysqlConnection** et **tMysqlCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Annuler l'insertion de données dans des tables mère/fille

En vous basant sur *Scénario 1 : Insérer des données dans des tables mère/fille* du composant **tMysqlConnection**, insérez une opération de rollback (annulation) afin d'éviter un commit inutile.

1. Glissez-déposez un **tMysqlRollback** de la **Palette** dans l'espace de modélisation graphique.
2. Connectez le **tFileList** au **tMysqlRollback** à l'aide d'un lien **OnSubjobError**.
3. Double-cliquez sur le **tMysqlRollback** afin d'ouvrir sa vue **Basic settings**.
4. Dans la liste **Component List**, sélectionnez le composant de connexion à utiliser.

Le **tMysqlRollback** assure que la transaction ne sera pas commitée partiellement.

tMysqlRow

Propriétés du tMysqlRow

Famille de composant	Databases/MySQL	
Fonction	Le **tMysqlRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tMysqlRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez la version de MySQL que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à traiter.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. *Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l'alias de la source de données spécifié.* Cette option est indisponible lorsque la case **Use an existing connection** est cochée.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre.

		Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Supprimer et re-générer un index de table MySQL

Le scénario suivant crée un Job à quatre composants permettant de supprimer un index de table, effectuer un Insert dans cette table avant de re-générer l'index.

Relier les composants

1. Sélectionnez les composants suivants dans la **Palette** et déposez-le dans l'espace de modélisation : deux **tMysqlRow**, un **tRowGenerator**et un **tMysqlOutput**.
2. Connectez le premier composant **tMysqlRow** au **tRowGenerator** à l'aide d'un lien **OnComponentOk**.
3. Reliez le **tRowGenerator** au **tMysqlOutput** à l'aide d'un lien **Row** > **Main**.

4. Connectez le composant **tRowGenerator** au second composant **tMysqlRow** à l'aide d'une connexion **OnSubjobOk**.

Configurer les composants

1. Sélectionnez le composant **tMysqlRow** et renseignez les propriétés de la base de données dans l'onglet **Basic settings**.

2. Dans les champs **Property type** et **Schema**, sélectionnez la base de données appropriée dans la liste.

   Les détails de connexion à la base de données ainsi que le schéma de la table sont renseignés automatiquement.

3. Propagez les informations de propriétés et de schéma aux autres composants du Job.

   La requête étant conservée dans les **Metadata** du Repository, vous pouvez également sélectionner **Repository** dans le champ **Query type** et cliquer sur la requête correspondante.

4. Si vous ne conservez pas vos requêtes dans le **Repository**, saisissez la déclaration SQL suivante : *drop index <index_name> on <table_name>*

5. Sélectionnez le deuxième composant **tMysqlRow**, vérifiez ses propriétés et son schéma.

6. Puis saisissez la déclaration SQL permettant de recréer un index à l'aide de la formulation suivante : *create index <index_name> on <table_name> (<column_name>);*

   Le composant **tRowGenerator** permet de générer automatiquement les colonnes à ajouter à la table de sortie définie.

7. Sélectionnez le composant **tMysqlOutput** et renseignez les propriétés de connexion à la base de données soit à partir du Repository ou manuellement si les informations de connexion ne sont utilisées que pour ce Job. La table à alimenter a pour nom : *comprehensive*.

8. Le schéma est automatiquement hérité du flux de données du **tLogRow**. Editez le schéma et vérifiez que sa structure correspond au schéma attendu par la table de base de données spécifiée.

   Dans le champ **Action on table**, sélectionnez *None* et dans le champ **Action on data**, sélectionnez *Insert*.

   Aucune colonne supplémentaire n'est requise pour ce Job.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegardez le Job.
2. Appuyez sur **F6** pour exécuter ce Job.

   Si vous avez la possibilité d'observer l'action sur la base de données, remarquez que l'index a été supprimé en début de Job puis recréé à la fin de l'action d'Insert.

   Voir également *tDBSQLRow*.

Scénario 2 : Utiliser l'instance PreparedStatement pour faire une requête sur des données

Le scénario suivant décrit un Job à quatre composants permettant de lier la colonne d'une table à un fichier clients. En effet, la table MySQL contient la liste de tous les Etats américains avec leur identifiant, et le fichier contient des informations sur vos clients avec l'identifiant de l'état dans lequel ils résident. L'objectif de ce scénario est de récupérer le nom de l'Etat pour chaque client à l'aide d'une requête SQL. Afin de traiter un grand nombre de données plus rapidement, utilisez l'objet PreparedStatement permettant de n'exécuter qu'une seule fois la requête au lieu de l'effectuer à chaque ligne. Chaque ligne de données est envoyée comme paramètre. Un objet PreparedStatement peut également être utilisé pour éviter une injection SQL.

Pour ce scénario, utilisez un fichier et une base de données dont vous avez préalablement stocké la connexion et les propriétés dans des métadonnées dans le **Repository**. Pour plus d'informations concernant la création de métadonnées de fichiers délimités, la création de métadonnées de connexion à une base de données et l'utilisation de ces métadonnées, consultez le *Guide utilisateur du Studio Talend*.

- A partir du **Repository**, développez le nœud **Metadata** et **File delimited**
- Sélectionnez la métadonnée correspondant au fichier clients que vous souhaitez utiliser dans ce Job. Ici, vous utilisez la métadonnée *customers*.
- Glissez cette métadonnée dans l'espace de modélisation et double-cliquez sur **tFileInputDelimited** dans la boîte de dialogue **[Components]** afin de créer un composant **tFileInputDelimited** déjà paramétré.

- Dans la liste **Schema**, sélectionnez **Built-in** afin de modifier le schéma de votre composant. Puis, cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ajouter la colonne dans laquelle insérer le nom de l'Etat.

Column	Key	Type	N..	Date Patt...	Length	Precision
id		Integer			2	
CustomerName		String			30	
CustomerAddress		String			25	
idState		Integer			2	
id2		Integer			2	
RegTime		String			16	
RegisterTime		String			23	
Sum1		Float			7	2
Sum2		Float			9	4
Sum3		Float			7	2
Sum4		Float			5	3
Sum5		Float			9	4
Sum6		Float			9	2
Sum7		String			12	
LabelStateRecordSet		Object				

- Cliquez sur le bouton **[+]** pour ajouter une colonne au schéma. Renommez cette colonne *LabelStateRecordSet*, et elle est de type objet, donc sélectionnez **Object** dans la liste **Type**. Cliquez sur **OK** pour enregistrer vos modifications.

- A partir de la **Palette**, sélectionnez les composants **tMysqlRow**, **tParseRecordSet** et **tFileOutputDelimited** et glissez-les dans l'espace de modélisation.

- Reliez les quatre composants via des liens de type **Row** > **Main**.

- Double-cliquez sur le composant **tMysqlRow** pour paramétrer ses propriétés dans l'onglet **Basic settings** de la vue **Component**.

- Dans la liste **Property Type**, sélectionnez **Repository** et cliquez sur le bouton **[...]** pour sélectionner la connexion à la base de données centralisée dans les métadonnées du Repository. Les champs **DB Version**, **Host**, **Port**, **Database**, **Username** et **Password** sont renseignés automatiquement. Si vous êtes en mode **Built-in**, renseignez ces champs manuellement.

- Dans la liste **Schema**, sélectionnez **Built-in** pour paramétrer le schéma manuellement et ajouter la colonne *LabelStateRecordSet*, ou cliquez directement sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

- Dans le champ **Query**, saisissez la requête SQL à utiliser. Ici, vous souhaitez récupérer les noms des Etats américains contenus dans la colonne *LabelState* de la table MySQL *us_state* : `"SELECT LabelState FROM us_state WHERE idState=?"`. Le point d'interrogation `"?"` représente le paramètre à définir dans l'onglet **Advanced settings**.

- Cliquez sur l'onglet **Advanced settings** pour paramétrer les propriétés avancées du composant.

- Cochez la case **Propagate QUERY's recordset** et sélectionnez la colonne *LabelStateRecordSet* dans la liste **use column** pour insérer le résultat de la requête dans cette colonne.
- Cochez la case **Use PreparedStatement** et définissez le paramètre utilisé dans la requête dans le tableau **Set PreparedStatement Parameters**.
- Cliquez sur le bouton **[+]** pour ajouter un paramètre.
- Dans la cellule **Parameter Index**, saisissez la position du paramètre dans l'instruction SQL. Saisissez "*1*" étant donné que vous n'utilisez qu'un seul paramètre dans l'exemple.
- Dans la cellule **Parameter Type**, saisissez le type du paramètre. Ici, le paramètre est de type entier, donc sélectionnez **Int** dans la liste.
- Dans la cellule **Parameter Value**, saisissez la valeur du paramètre. Ici, vous souhaitez récupérer le nom de l'Etat en fonction de leur ID pour chaque client du fichier d'entrée, saisissez donc `"row1.idState"`.
- Double-cliquez sur le composant **tParseRecordSet** pour paramétrer ses propriétés dans l'onglet **Basic settings** de la vue **Component**.

- Dans la liste **Prev. Comp. Column list**, sélectionnez la colonne du composant précédent à analyser. Dans cet exemple, sélectionnez la colonne *LabelStateRecordSet*.
- Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent et le tableau **Attribute table** est automatiquement renseigné avec les colonnes du schéma.
- Dans le tableau **Attribute table**, dans le champ **Value** correspondant à la colonne *LabelStateRecordSet*, saisissez entre guillemets le nom de la colonne contenant le nom des Etats à récupérer et à mettre en correspondance avec chaque client. Dans cet exemple, saisissez "*LabelState*".
- Double-cliquez sur le composant **tFileOutputDelimited** pour paramétrer ses propriétés dans l'onglet **Basic settings** de la vue **Component**.

- Dans le champ **File Name**, renseignez le chemin d'accès et le nom du fichier de sortie.
- Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

customers.csv

	A	B	C	D	E	F	G	H	O
1	1	Griffith Paving	talend@apres	7	41	03/11/2006 09:20	2001-01-17 06:	67852.0	Connecticut
2	2	Bill's Dive Sh	511 Maple Av	35	5	19/11/2004 15:48	2002-06-07 09:	88792.0	Ohio
3	3	Childress Chi	662 Lyons Ci	1	28	16/02/2005 08:27	1990-04-01 21:	35340.0	Alabama
4	4	Facelift Kitch	220 Vine Ave	41	15	22/08/2002 09:55	1972-04-23 18:	6097.0	South Dakota
5	5	Terrinni & So	770 Exmoor	5	9	28/06/2001 09:15	1982-04-19 10:	5146.0	California
6	6	Kermit the Pe	1860 Parksid	28	15	17/08/2003 10:07	2006-05-27 17:	16087.0	Nevada
7	7	Tub's Furnitur	807 Old Trail	15	9	27/08/2000 03:13	1970-03-27 23:	53216.0	Iowa
8	8	Toggle & Mye	618 Sheriden	9	15	24/03/2006 23:07	2005-08-02 01:	74168.0	Florida
9	9	Childress Chi	788 Tennyso	12	33	10/09/2001 06:33	1994-05-03 11:	92176.0	Idaho
10	10	Elle Hypnosis	2032 Northbr	1	7	11/01/1977 03:07	1975-06-10 20:	48498.0	Alabama

Une colonne contenant le nom de l'Etat américain correspondant à chaque client a été ajoutée au fichier.

Scénario 3 : Combiner deux flux pour une sortie sélective

Dans ce scénario, un flux généré par un **tFixedFlowInput** est combiné avec un flux de la base de données MySQL. Le flux source contient les champs *id* et *age* alors que la table MySQL contient *id* et *name*. Vous allez récupérer les données *age* du flux source et les combiner avec les enregistrements *id* et *name* de la table MySQL à partir de la mise en correspondance avec *id*. Le schéma d'entrée est différent de celui de sortie, dans le **tMysqlRow**.

Relier les composants

1. Déposez un **tFixedFlowInput**, un **tMysqlRow**, un **tParseRecordSet** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.

2. Renommez le **tFixedFlowInput** en **source_flow**, le **tMysqlRow** en **insert_recordset**, le **tParseRecordSet** en **parse_recordset** et le **tLogRow** en **show_combined_flow**.

3. Reliez le **tFixedFlowInput** au **tMysqlRow** à l'aide d'un lien **Row** > **Main**.

4. Reliez le **tMysqlRow** au **tParseRecordSet** à l'aide d'un lien **Row** > **Main**.

5. Connectez le **tParseRecordSet** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Sélectionnez l'option **Use Inline Content (delimited file)** dans la zone **Mode**.

   Dans le champ **Content**, saisissez les données à transférer :

```
1;30
2;20
```

3. Double-cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

   Cliquez sur le bouton **[+]** pour ajouter deux colonnes, nommées *id* et *age*, de type *Integer*.

   Cliquez sur **OK** pour fermer l'éditeur.

4. Double-cliquez sur le **tMysqlRow** pour ouvrir sa vue **Basic settings**.

5. Dans les champs **Host** et **Port**, saisissez les informations de connexion.

   Dans le champ **Database**, saisissez le nom de la base de données.

   Dans les champs **Username** et **Password**, saisissez les informations d'authentification.

   Dans le champ **Query**, saisissez la requête SQL permettant de récupérer les données de *id* et *name* dans la table MySQL *employee* : `"select id, name from employee WHERE id=?"`.

   Le point d'interrogation, "?" représente le paramètre à configurer l'onglet **Advanced settings**.

6. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

7. Cliquez sur le bouton **[+]** pour ajouter deux colonnes, à droite, nommées *recordset* et *age*, de type *Object* et *Integer*. Notez que *recordset* doit contenir les résultats de la requête de la table MySQL, c'est-à-dire les champs *id* et *name*.

   Cliquez sur **OK** pour fermer l'éditeur.

8. Cliquez sur l'onglet **Advanced settings** afin de configurer les paramètres avancés.

9. Cochez la case **Propagate QUERY's recordset** et sélectionnez *recordset* dans la liste **use column** afin d'insérer les résultats de la requête dans cette colonne.

   Cochez la case **Use PreparedStatement** et définissez les paramètres utilisés dans la requête, dans la table **Set PreparedStatement Parameters**.

10. Cliquez sur le bouton **[+]** pour ajouter une ligne.

    Dans la cellule **Parameter Index**, saisissez la position du paramètre dans l'instruction SQL. Saisissez "*1*" puisque vous utilisez un paramètre dans cet exemple.

    Dans la cellule **Parameter Type**, saisissez le type de paramètre. Le paramètre est de type *Integer*. Sélectionnez **Int** dans la liste.

    Dans la cellule **Parameter Value**, saisissez la valeur du paramètre. Ici, vous allez récupérer les colonnes *id* et *name* de la table *employee* selon la valeur de *id* du flux source. Saisissez *row3.id*.

11. Double-cliquez sur le composant **tParseRecordSet** pour ouvrir sa vue **Basic settings**.

    Dans la liste **Prev. Comp. Column list**, sélectionnez la colonne à analyser, *recordset*.

12. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

Cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes, à droite, nommées *id*, *name* et *age*, respectivement de type *Integer*, *String* et *Integer*. Les colonnes *id* et *name* doivent contenir les données analysées de *recordset*.

Cliquez sur **OK** pour fermer l'éditeur.

Dans la table **Attribute table**, dans les champs **Value** correspondants aux colonnes *id* et *name*, saisissez le nom des colonnes de la table MySQL à récupérer, *"id"* et *"name"*.

13. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage sous forme de tableau.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** pour exécuter le Job.

```
[statistics] connecting to socket on port 4030
[statistics] connected
.--+--------+---.
|show_combined_flow|
|=-+--------+--=|
|id|name    |age|
|=-+--------+--=|
|1 |Andy    |30 |
|2 |Anderson|20 |
'--+--------+---'

[statistics] disconnected
```

tMysqlSCD

Le composant **tMysqlSCD** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tMysqlSCD*.

tMysqlSCDELT

Le composant **tMysqlSCDELT** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tMysqlSCDELT*.

tMysqlSP

Propriétés du tMysqlSP

Familles de composant	Databases/Mysql	
Fonction	Le composant **tMysqlSP** appelle une procédure stockée de base de données.	
Objectif	Le **tMysqlSP** permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier des propriétés du composant. Les champs suivants sont alors pré-remplis.
	Use an existing connection	Cochez cette case lorsque vous utilisez le composant **tMysqlConnection**.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification sur l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

	SP Name	Saisissez le nom exact de la Procédure Stockée (SP).
	Is Function / Return result in	Cochez cette case, si une seule valeur doit être retournée. Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.
	Parameters	Cliquez sur le bouton **[+]** et sélectionnez dans le champ **Schema Columns** les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure. Sélectionnez le **Type** de paramètre : **IN** : paramètre d'entrée (Input). **OUT** : paramètre de sortie (Output)/valeur retournée. **IN OUT** : les paramètres d'entrée doivent être retournés sous forme de valeur, même après modifications via la procédure (fonction). **RECORDSET** : les paramètres d'entrée doivent être retournés sous forme d'ensemble de valeurs, au lieu d'une valeur unique. Consultez *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire. Il peut également être utilisé comme composant de début. Dans ce cas, seuls les paramètres d'entrée sont autorisés.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données.	

Scénario : Trouver le libellé State à l'aide d'une procédure stockée

Le Job suivant a pour but de trouver les libellés *State* en fonction des ID impairs de la colonne *State* d'une table MySQL à deux colonnes à l'aide d'une procédure stockée.

- Cliquez et déposez les composants suivants dans l'espace de modélisation : **tRowGenerator**, **tMysqlSP** et **tLogRow**.
- Connectez les composants à l'aide d'une connexion de type **Row Main**.
- Le **tRowGenerator** est utilisé pour générer des ID impairs. Double-cliquez sur le composant pour lancer l'éditeur.

- Cliquez sur le bouton **[+]** pour ajouter une colonne au schéma à générer.
- Cochez la case **Key** pour définir l'entrée comme étant une clé primaire et définissez le **Type** en **Int** (entier).
- Dans le champ **Length**, la longueur est de 2 chiffres au maximum.
- Utilisez la fonction préétablie appelée **sequence** mais modifiez les paramètres dans la partie inférieure de la fenêtre.

- Modifiez la valeur (**Value**) de **step** : de *1* à *2* pour cet exemple. Mais la valeur de départ (start value) reste *1*.
- Dans le champ **Number of generated rows**, définissez le nombre de lignes à générer à *25*, afin que tous les ID impairs de *State* (il y a 50 Etats) soient générés.

- Cliquez sur **OK** pour valider la configuration.
- Puis sélectionnez le composant **tMysqlSP** et paramétrez ses propriétés.

- Dans le champ **Property type**, sélectionnez l'option **Repository** puis sélectionnez l'entrée correspondante dans la liste. Les informations de connexion sont remplies automatiquement.
- Sinon, paramétrez les informations de connexion manuellement.
- Cliquez sur **Sync Column** pour récupérer le schéma généré du composant précédent.
- Puis cliquez sur **Edit Schema** et ajoutez une colonne supplémentaire contenant les libellés des Etats (State) à obtenir en sortie, en plus de l'ID.
- Sélectionnez le type d'encodage dans la liste.
- Dans le champ **SP Name**, saisissez le nom de la procédure comme il a été défini dans la base de données (ici, *getstate*). La procédure à exécuter correspond à :

```
DROP PROCEDURE
IF EXISTS `talend`.`getstate` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getstate`(IN pid INT, OUT
pstate VARCHAR(50))
BEGIN
 SELECT LabelState INTO pstate FROM us_states WHERE idState = pid;
END $$
```

- Dans la zone **Parameters**, cliquez sur le bouton **[+]** pour ajouter une ligne à la table.
- Définissez le champ **Column** en *ID*, et le champ **Type** en *IN*, ainsi il sera considéré comme paramètre d'entrée dans la procédure.
- Ajoutez une deuxième ligne et définissez le champ **Column** en *State* et le champ **Type** en *Out*, ainsi il sera considéré comme le paramètre de sortie a retournée.
- Et enfin, paramétrez les propriétés du composant **tLogRow**.

- Synchronisez le schéma avec celui du composant précédent en cliquant sur le bouton **Sync columns**.
- Et cochez la case **Print values in cells of a table** pour un meilleur confort de lecture.
- Puis enregistrez votre Job et exécutez-le.

```
Starting job MysqlSP at 17:24 23/08/2007.
.--+--------------.
|     tLogRow_1    |
|=--+--------------=|
|ID|State          |
|=--+--------------=|
|1 |Alabama        |
|3 |Arizona        |
|5 |California     |
|7 |Connecticut    |
|9 |Florida        |
|11|Hawaii         |
|13|Illinois       |
|15|Iowa           |
|17|Kentucky       |
|19|Maine          |
|21|Massachusetts  |
|23|Minnesota      |
|25|Missouri       |
|27|Nebraska       |
|29|New Hampshire  |
|31|New Mexico     |
|33|North Carolina|
```

La sortie affiche les différents Etats (State) avec leur ID impair comme défini dans la procédure.

Consultez également *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario : Récupérer des informations personnelles à l'aide d'une procédure stockée*.
- *Scénario : Vérifier le format de numéros à l'aide d'une procédure stockée*.
- *Scénario : Exécuter une procédure stockée dans le hub MDM*.

tMysqlTableList

Propriétés du tMysqlTableList

Famille de composant	Databases/MySQL	
Fonction	Le composant **tMysqlTableList** effectue une opération d'itération sur toutes les tables d'une base de données, grâce à une connexion MySQL définie.	
Objectif	Le composant **tMysqlTableList** permet de lister les noms des tables MySQL grâce aux commandes SELECT et WHERE.	
Basic settings	*Component list*	Sélectionnez le composant **tMysqlConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Where clause for table name selection	Saisissez la commande WHERE permettant d'identifier les tables sur lesquelles effectuer l'opération d'itération.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**CURRENT_TABLE** : nom de la table sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_TABLE** : nombre de tables itérées jusqu'à présent. Cette variable est une variable **Flow** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Vous pouvez utiliser ce composant en association avec les autres composants MySQL, notamment avec le **tMysqlConnection**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Effectuer une opération d'itération sur des tables et en effacer le contenu grâce à un modèle SQL défini par l'utilisateur (SQL Template)

Le scénario Java suivant décrit un Job à trois composants qui permet d'effectuer une opération d'itération, dans les tables spécifiées d'une base de données MySQL, en utilisant une clause de type WHERE. Le Job permet ensuite d'effacer le contenu de ces tables directement sur le SGBD grâce à un modèle SQL (SQL Template) défini par l'utilisateur.

Pour une utilisation avancée, commencez par créer une connexion à la base de données qui contient les tables que vous voulez vider de leur contenu.

- Dans la vue en arborescence du **Repository**, développez le nœud **Metadata** puis cliquez-droit sur **DB Connections** pour créer une connexion à la base de données de travail et stocker les informations de connexion en local.

  Pour plus d'informations sur les métadonnées (metadata), consultez le *Guide utilisateur du Studio Talend*.

  Sinon, déposez un composant **tMysqlConnection** dans l'éditeur graphique et renseignez les paramètres de connexion manuellement.

- A partir du **Repository**, déposez dans l'éditeur graphique la connexion que vous venez de créer.

  La boîte de dialogue **[Components]** s'affiche à l'écran.

- Sélectionnez le composant **tMysqlConnection** puis cliquez sur **OK**.

  Le composant **tMysqlConnection** apparait alors dans l'éditeur graphique et les paramètres de connexion, accessibles depuis la vue **Basic settings**, sont automatiquement renseignés.

- A partir de la **Palette**, cliquez-déposez les composants **tMysqlTableList** et **tELT** dans l'éditeur graphique.

- Reliez les composants **tMysqlConnection** et **tMysqlTableList** à l'aide d'un lien Trigger de type **OnSubjobOk**.

- Reliez les composants **tMysqlTableList** et **tELT** à l'aide d'un lien de type **Iterate**.

- Double-cliquez sur le **tMysqlConnection** pour afficher la vue **Basic settings** du composant si vous voulez en vérifier les paramètres de connexion.

Dans cet exemple, connectez-vous à une base de données MySQL appelée *examples*.

- Dans l'éditeur graphique, double-cliquez sur le composant **tMysqlTableList** pour en afficher les paramètres de base et les définir (vue **Basic settings**).

- Sélectionnez la connexion MySQL appropriée à partir de la liste déroulante **Component list**, dans le cas où vous utilisez plus d'une connexion.
- Renseignez la clause de type WHERE dans le champ approprié, en veillant à utiliser la bonne syntaxe, pour effectuer une itération sur la ou les tables dont vous voulez effacer le contenu.

  Dans ce scénario, opérez l'itération sur toutes les tables dont le nom commence par "ex".

- Dans l'éditeur graphique, double-cliquez sur le composant **tELT** pour en afficher les paramètres de base et les définir (vue **Basic settings**).

- Dans le champ **Database Name**, renseignez le nom de la base de données où se trouvent les tables sur lesquelles vous voulez travailler.
- Sélectionnez la connexion MySQL appropriée à partir de la liste déroulante **Component list**, dans le cas où vous utilisez plus d'une connexion.
- Cliquez dans le champ **Table name** et appuyez sur **Ctrl+Espace** pour atteindre la liste générale des variables.
- A partir de cette liste de variables, sélectionnez *((String)globalMap.get("tMysqlTableList_1_CURRENT_TABLE"))*.

Comment créer votre propre modèle SQL (SQL Template) :

- Dans la vue en arborescence du **Repository**, développez successivement les nœuds **SQL Templates** et **MySQL**.

- Cliquez-droit sur le nœud **UserDefined** et sélectionnez l'option **Create SQLTemplate** à partir de la liste.

  L'assistant **[New SQLTemplate]**s'affiche à l'écran.

- Saisissez un nom pour le nouveau modèle (SQL template) et remplissez les autres champs si nécessaire, puis cliquez sur **Finish** pour fermer l'assistant.

  L'éditeur de modèle SQL s'affiche dans l'éditeur graphique.

- Supprimez le code existant et saisissez le code permettant d'exécuter l'action voulue, c'est-à-dire, ici, effacer le contenu de toutes les tables dont les noms commencent par "ex".

Dans un code SQL template, il est impératif d'utiliser le nom exact de la variable, en association avec le paramètre désignant le nom de la table ("__TABLE-NAME__" dans cet exemple). Pour afficher le nom de la variable utilisée, positionnez votre curseur sur le champ **Table Name** de la vue **Basic settings** du composant **tELT**.

- Appuyez sur **Ctrl+S** pour sauvegarder le nouveau modèle que vous venez de définir.

La prochaine étape consiste à ajouter ce nouveau modèle SQL template à la liste de modèles du même type dans le composant **tELT**.

Comment ajouter votre propre modèle SQL à la liste de modèles du même type :

- Dans la vue **Component** du composant **tELT**, cliquez sur l'onglet **SQL Templates** pour afficher la liste de modèles **SQLTemplate List**.

- Cliquez sur le bouton **Add** et ajoutez deux lignes de modèles SQL template.
- Cliquez sur la première ligne pour afficher un menu déroulante puis cliquez sur la flèche de ce menu pour dérouler la liste des modèles.

- Dans la liste, sélectionnez le modèle SQL template que vous venez de créer.
- Assurez-vous que le modèle de la seconde ligne est bien en mode **Commit**.
- Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

Toutes les tables de la base de données MySQL `examples` dont le nom commence par "ex" sont alors vidées de leur contenu.

Scénario associé

Pour un scénario associé, consultez *Scénario : Itérer une table de base de données et lister le nom des colonnes de la table*.

tOleDbInput

Propriétés du tOleDbInput

Famille de composant	Databases/OleDb	
Fonction	Ce composant lit une base de données et en extrait des champs à l'aide de requêtes. Il offre la flexibilité de la requête en base de données et couvre toutes les possibilités de requête SQL.	
Objectif	Le **tMysqlInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Connection URL*	Renseignez la chaîne de connexion utilisée pour ouvrir la base de données. Pour plus de détails, consultez http://msdn.microsoft.com/fr-fr/library/system.data.oledb.oledbconnection.connectionstring.aspx.
	Schema and *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Saisissez le nom de table à traiter.
	Query Type	Sélectionnez le type de requête **Built-in** ou **Repository** dans la liste déroulante. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de la table définie dans le champ **Query**.
	Guess schema	Cliquez sur le bouton **Guess schema** pour récupérer le schéma de la table.
	Query	Spécifiez la requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Trim all the String/Char columns*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

	Trim column	Cochez cette case dans la colonne **Trim** pour supprimer les espaces en début et en fin de champ dans les colonnes correspondantes. Ce champ apparaît uniquement lorsque la case **Trim all the String/Char columns** est décochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé en tant que composant d'entrée d'un Job et nécessite un lien de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario similaire, consultez le *tMysqlInput*.

tOleDbOutput

Propriétés du tOleDbOutput

Famille de composant	Databases/OleDb	
Fonction	Ce composant écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Ce composant exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Database*	Renseignez la chaîne de connexion utilisée pour ouvrir la base de données. Pour plus de détails, consultez http://msdn.microsoft.com/fr-fr/library/system.data.oledb.oledbconnection.connectionstring.aspx.
	Table	Spécifiez le nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on data	Dans la liste déroulante, sélectionnez parmi les opérations suivantes celle que vous souhaitez effectuer sur les données. • **Insert** : insérer des données. • **Update** : mettre à jour des données. • **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. • **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. • **Delete** : supprimer des données. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations Update et Delete. Pour cela, cliquez sur le bouton [...] à côté du champ Edit Schema et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet Advanced settings pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case Use field options et cochez la case Key in update correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases Key in delete pour les opérations de suppression (Delete).*
	Clear data in table	Cochez cette case pour supprimer les données dans la table définie avant toute opération.
	Schema and *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Additional columns*	Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier. • **Name** : Saisissez le nom de la colonne à modifier ou à insérer. • **SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer des données dans les colonnes correspondantes. • **Position** : Sélectionnez **Before**, **After** or **Replace**, en fonction de l'action à effectuer sur la colonne de référence. • **Reference column** : Saisissez une colonne de référence utilisée pour situer ou remplacer une nouvelle colonne ou celle à modifier. Cette option n'est pas disponible si vous venez de créer la table de base de données (même si vous la supprimez avant).
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Field options	Cochez cette ou ces case(s) dans la table si vous souhaitez utiliser la ou les colonne(s) de schéma correspondantes en tant que base pour votre opération. • **Key in update** : définir la clé primaire à utiliser lors d'une opération de mise à jour. • **Key in delete** : définir la clé primaire à utiliser lors d'une opération de suppression. • **Updatable** : définir la ou les colonne(s) à mettre à jour. • **Insertable** : définir la ou les colonne(s) à insérer. Ce champ apparaît uniquement lorsque la case **Use field options** est cochée.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est normalement utilisé en tant que composant de sortie d'un Job et nécessite un lien d'entrée.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario similaire, consultez le *tMysqlOutput*.

tOleDbRow

Propriétés du tOleDbRow

Famille de composant	Databases/OleDb	
Fonction	**tOleDbRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée.	
Objectif	Selon la nature de la requête et de la base de données, le **tOleDbRow** agit sur la structure même de la base de données ou sur les données.	
Basic settings	*Database*	Saisissez la chaîne de caractères permettant d'ouvrir la base de données. Pour plus d'informations, consultez la page http://msdn.microsoft.com/fr-fr/library/system.data.oledb.oledbconnection.connectionstring.aspx.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à traiter.
	Query type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.

		Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario similaire, consultez *tMysqlRow*.

tOracleBulkExec

Propriétés du tOracleBulkExec

Les composants **tOracleOutputBulk** et **tOracleBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tOracleOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Famille de composant	Databases/Oracle	
Fonction	Le composant **tOracleBulkExec** insère, ajoute, remplace ou supprime les données d'une base de données Oracle.	
Objectif	Le **tOracleBulkExec** est un composant dédié, il permet un gain de performance lors du traitement des données de la base de données Oracle.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Liste déroulante des pilotes disponibles: **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle. **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante. **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.

	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Data file name	Nom du fichier à charger. *Ce fichier doit être situé sur la même machine que le serveur de base de données.*
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Insérer des lignes dans une table vide. Le Job s'arrête en cas de doublons. **Update** : Mettre à jour les entrées existantes. **Append** : Ajouter des lignes aux données déjà existantes. **Replace** : Ecraser des lignes de la table. **Truncate** : Supprimer les entrées correspondant au flux d'entrée. *Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations d'Update. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (**Advanced settings**) pour une utilisation optimisée de ces opérations.*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Use existing control file	Cochez cette case si vous utilisez un fichier de contrôle (.ctl) et spécifiez son chemin d'accès dans le champ **.ctl file name**.
	Record format	Définissez le format de l'enregistrement : **Default** : les paramètres du format sont définis par défaut. **Stream** : déterminez le séparateur de fin d'enregistrement. **Fixed** : déterminez la longueur d'enregistrement. **Variable** : déterminez la taille du champ spécifiant la longueur de l'enregistrement.
	Specify .ctl file's INTO TABLE clause manually	Cochez cette case pour renseigner manuellement la clause INTO TABLE du fichier de contrôle, directement dans le code.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs : **None** : aucun caractère de séparation. **Whitespace** : le caractère de séparation est un espace. **EOF (used for loading LOBs from lobfile)** : la séparation est constituée d'un caractère de fin de fichier (End Of File). **Other terminator** : la séparation est constituée d'un caractère autre que ceux précédemment énoncés.
	Use fields enclosure	Cochez cette case si vous voulez utiliser des caractères pour entourer le texte : **Fields enclosure (left part)** : caractère délimitant les champs sur la gauche. **Field enclosure (right part)** : caractère délimitant les champs sur la droite.
	Use schema's Date Pattern to load Date field	Cochez cette case pour utiliser le modèle de date contenu dans le schéma pour remplir le champ concernant la date.
	Specify field condition	Cochez cette case pour définir une condition de chargement des données.
	Preserve blanks	Cochez cette case pour conserver les espaces vides.
	Trailing null columns	Cochez cette case pour charger des données avec des colonnes vides.
	Load options	Cliquez sur le bouton **[+]** afin d'ajouter des options de chargement des données : **Parameter** : choisissez l'un des paramètres de chargement à partir de la liste déroulante. **Value** : définissez la valeur à attribuer au paramètre.
	NLS Language	A partir de la liste déroulante, définissez la langue utilisée pour les données qui ne sont pas codées en Unicode.
	Set Parameter NLS_TERRITORY	Cochez cette case pour modifier les conventions d'appellation utilisées pour la numérotation des jours et des semaines. La valeur par défaut est celle du système d'exploitation.
	Encoding	Sélectionnez l'encodage dans la liste ou saisissez-le entre guillemets doubles si le type d'encodage utilisé n'existe pas dans la liste. Ce champ est obligatoire pour la gestion de données de bases de données.
	Output	Sélectionnez la sortie à utiliser pour récupérer la sortie standard de la base Oracle : **to console** : vers la console.

		to global variable : vers la variable globale.
	Convert columns and table names to uppercase	Cochez cette case pour mettre le nom des colonnes et celui de la table en majuscules.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_DATA** : nombre de lignes lues. Cette variable est une variable **After** et retourne un entier. **NB_LINE_BAD** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **RETURN_CODE** : code retour indiquant le résultat du traitement. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant dédié offre performance et flexibilité lors de requêtes sur les bases de données Oracle.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Le serveur/client de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tOracleBulkExec** afin que le composant fonctionne correctement.	

Scénario : Supprimer et insérer des données dans une base Oracle

Ce scénario décrit comment supprimer le contenu d'une base de données Oracle et insérer de nouvelles données à partir du contenu d'un fichier d'entrée. Le Job est constitué de trois composants qui créent le contenu, le transfèrent dans un fichier qui sera ensuite chargé dans la base Oracle après suppression des données de la table.

Construire le Job

1. Cliquez et déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tOracleInput**, **tFileOutputDelimited** et **tOracleBulkExec**.

2. Connectez le composant **tOracleInput** au **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main**.

3. Reliez le **tOracleInput** au **tOracleBulkExec** à l'aide d'un lien **OnSubjobOk**.

Configurer les composants

1. Paramétrez les informations de connexion à la base de données Oracle. Il est conseillé de stocker ces informations dans une métadonnée du **Repository** pour pouvoir les réutiliser par la suite dans tous vos Jobs.

2. Paramétrez le schéma, s'il n'est pas conservé dans le **Repository**. Dans cet exemple, le schéma contient les quatre colonnes suivantes : *ID_Contract*, *ID_Client*, *Contract_type* et *Contract_Value*.

3. Dans la liste **Encoding**, sélectionnez l'option **Custom** et définissez *AL32UTF8* comme encodage.

4. Dans la vue **Basic settings** du composant **tFileOutputDelimited**, paramétrez ses propriétés, notamment les champs **File Name**, **Row separator** et **Field separator** avec respectivement le chemin d'accès et le nom du fichier, le séparateur de lignes et de champs.

5. Double-cliquez ensuite sur le composant **tOracleBulkExec** pour définir les propriétés d'alimentation de la base de données.

6. Dans la liste **Property Type**, sélectionnez le mode **Repository** si vous avez stocké les informations de connexion à la base de données sous le nœud **Metadata** du **Repository** ou sélectionnez le mode **Built-In** pour les définir manuellement. Dans ce scénario, utilisez le mode **Built-In**.

7. Configurez si nécessaire les paramètres de connexion dans les champs **Host**, **Port**, **Database**, **Schema**, **Username** et **Password**.

8. Dans le champ **Table**, renseignez le nom de la table à alimenter et dans le champ **Action on data**, sélectionnez l'action à effectuer, ici, **Insert**.

9. Dans la liste **Schema**, sélectionnez le mode **Built-In** puis cliquez sur le bouton **[...]** à côté du champ **Edit Schema** pour définir les données à passer au composant suivant.

10. Cliquez sur l'onglet **Advanced settings** pour configurer les paramètres avancés du composant.

11. Cochez la case **Use an existing control file** si vous avez stocké les informations relatives à l'état de la base de données dans un fichier de contrôle (.ctl). Sinon, renseignez, si nécessaire, les champs suivants manuellement : **Record format**, **Specify .ctl file's INTO TABLE clause manually**, **Field terminated by**, **Use field enclosure**, **Use schema's Date Pattern to load Date field**, **Specify field condition**, **Preserve blanks**, **Trailing null columns**, **Load options**, **NLS Language** et **Set Parameter NLS_TERRITORY** en fonction de votre base de données.

12. Dans la liste **Encoding**, sélectionnez le type d'encodage ou saisissez-le entre guillemets doubles s'il n'est pas dans la liste.

13. Dans la liste **Output**, sélectionnez l'option **to console**, pour afficher la sortie standard de la base dans la console.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** pour exécuter le Job. Le log s'affiche dans la console de la vue **Run** et la table est alimentée par les données du fichier paramétré.

Voir également : *Scénario : Insérer des données dans une base MySQL*.

tOracleClose

Propriétés du tOracleClose

Famille de composant	Databases/Oracle	
Fonction	Le composant **tOracleClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Oracle.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tOracleConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Oracle, notamment **tOracleConnection** et **tOracleCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tOracleConnection** et **tOracleRollback**. Il est généralement utilisé avec un composant **tOracleConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tOracleClose**, consultez *tMysqlConnection*.

tOracleCommit

Propriétés du tOracleCommit

Famille de composant	Databases/Oracle	
Fonction	Le composant **tOracleCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tOracleConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** >**Main** pour relier le **tOracleCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Oracle et notamment **tOracleConnection** et **tOracleRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tOracleCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tOracleConnection

Propriétés du tOracleConnection

Famille de composant	Databases/Oracle	
Fonction	Le composant **tOracleConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Connection type	Liste déroulante des pilotes disponibles: **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle. **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante. **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.
	Use tns file	Cochez cette case pour utiliser les métadonnées d'un contexte contenu dans un fichier Tns. Un même fichier Tns peut contenir plusieurs contextes. **TNS File** : Renseignez manuellement le chemin d'accès ou cliquez sur le bouton **[...]** pour parcourir vos dossiers jusqu'au fichier Tns à utiliser. **Select a DB Connection in Tns File** : Cliquez sur le bouton **[...]** pour afficher tous les contextes du fichier Tns et sélectionner celui à utiliser.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le

		mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Additional JDBC parameters	Spécifiez des informations supplémentaires de connexion à la base de données créée. Dans ce champ, vous pouvez paramétrer le type d'encodage.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants Oracle, notamment les composants **tOracleCommit** et **tOracleRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tOracleConnection**, consultez *tMysqlConnection*.

tOracleInput

Propriétés du tOracleInput

Famille de composant	Databases/ Oracle	
Fonction	Le composant **tOracleInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tOracleInput** exécute une requête de base de données dans laquelle l'ordre doit strictement correspondre à l'ordre défini dans le schéma. Puis la liste des champs est transmise au composant via une connexion Main Row.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionne le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Liste déroulante des pilotes disponibles: **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle. **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante. **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.

	Oracle schema	Nom du schéma Oracle.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Table name	Nom de la table de base de données.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. *Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l'alias de la source de données spécifié.* Cette option est indisponible lorsque la case **Use an existing connection** est cochée.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
	Use cursor	Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d'optimiser les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	No null values	Cochez cette case pour optimiser les performances s'il n'y a pas de valeur nulle.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	

Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Oracle.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Utiliser les paramètres de contexte lors de la lecture d'une table d'une base de données Oracle

Dans ce scénario, une table est lue dans une base de données Oracle, à l'aide d'un paramètre de contexte pour référer au nom de la table.

Déposer et relier les composants

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l'espace de modélisation graphique et en les déposant depuis la **Palette** : un **tOracleInput** et un **tLogRow**.

2. Reliez le **tOracleInput** au **tLogRow** via un lien **Row > Main**.

Configurer les composants

1. Double-cliquez sur le **tOracleInput** pour ouvrir sa vue **Basic settings**.

2. Sélectionnez une connexion dans la liste déroulante **Connection Type**. Dans cet exemple, saisissez *OracleSID*.

   Sélectionnez la version de la base de données Oracle à utiliser dans la liste déroulante **DB Version**. Dans cet exemple, saisissez *Oracle 12-7*.

   Dans le champ **Host**, saisissez l'adresse IP du serveur de la base de données Oracle, *"192.168.31.32"*.

   Dans le champ **Database**, saisissez le nom de la base de données, *"TALEND"* dans cet exemple.

   Dans le champ **Oracle schema**, saisissez le nom du schéma Oracle, *"TALEND"* dans cet exemple.

   Dans les champs **Username** et **Password**, saisissez les informations d'authentification.

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

4. Cliquez sur le bouton pour ajouter quatre colonnes : *ID* et *AGE* de type **Integer** et *NAME* et *SEX*, de type **String**.

   Cliquez sur **OK** pour fermer l'éditeur du schéma et acceptez la propagation proposée par la boîte de dialogue qui s'ouvre.

5. Placez votre souris sur le champ **Table Name** et appuyez sur **F5** pour configurer les paramètres de contexte. La boîte de dialogue **[New Context Parameter]** s'ouvre.

   Pour plus d'informations concernant les paramètres de contexte, consultez le *Guide utilisateur du Studio Talend*.

6. Dans le champ **Name**, laissez le nom du paramètre de contexte, *TABLE* dans cet exemple.

   Dans le champ **Default value**, saisissez le nom de la table de base de données Oracle à interroger. Dans cet exemple, saisissez *PERSON*.

7. Clique sur **Finish** pour valider.

   La paramètre de contexte *context.TABLE* apparaît automatiquement dans le champ **Table Name**.

8. Dans la liste **Query Type**, sélectionnez **Built-In**. Cliquez sur **Guess Query** pour obtenir l'instruction de la requête.

```
"SELECT
  "+context.TABLE+".\"ID\",
  "+context.TABLE+".NAME,
  "+context.TABLE+".SEX,
  "+context.TABLE+".AGE
FROM "+context.TABLE
```

9. Double-cliquez sur le **tLogRow** afin d'ouvrir sa vue **Basic settings**.

10. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un meilleur affichage des résultats.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter le Job.

```
[statistics] connecting to socket on port 3750
[statistics] connected
.--+-------+------+---.
|       tLogRow_1      |
|=-+-------+------+--=|
|ID|NAME   |SEX   |AGE|
|=-+-------+------+--=|
|1 |Ford   |Male  |25 |
|2 |Rose   |Female|30 |
|3 |Sabrina|Female|28 |
|4 |Teddy  |Male  |32 |
|5 |Kate   |Male  |35 |
'--+-------+------+---'

[statistics] disconnected
```

Comme affiché dans la capture d'écran, les données dans la table de la base de données Oracle *PERSON* sont affichées dans la console.

Scénarios associés

Consultez les scénarios associés :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table* du composant **tDBInput**.
- *Scénario 2 : Utiliser la variable StoreSQLQuery* du composant **tDBInput**.
- *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tOracleOutput

Propriétés du tOracleOutput

Famille de composant	Databases/Oracle	
Fonction	Le composant **tOracleOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tOracleOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Liste déroulante des pilotes disponibles: **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle. **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante. **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.

	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	La liste **Action on table** n'est pas disponible si vous cochez la case **Enable parallel execution** dans la vue **Advanced settings**. Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **Default** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table. **Truncate table** : supprimer le contenu de la table, mais sans possibilité de Rollback. **Truncate table with reuse storage** : supprimer le contenu de la table. Il n'y a pas de possibilité de rollback. Cependant, vous pouvez réutiliser le stockage alloué à la table, même si le stockage est considéré comme vide. *Si vous cochez la case **Use an existing connection** et que vous sélectionnez une option différente de **Default** dans la liste **Action on table**, une instruction de commit est générée automatiquement, avant l'opération d'insertion/mise à jour/suppression de données.*
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondant au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*

	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. *Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l'alias de la source de données spécifié.* Cette option est indisponible lorsque la case **Use an existing connection** est cochée.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Override any existing NLS_LANG environment variable	Cochez cette case pour écraser les variables d'environnement NLS_LANG déjà définies.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet

		d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tOracleOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Use Hint Options	Cochez cette case pour ajouter des indicateurs (ou Hints) permettant d'optimiser le plan d'exécution de la requête afin d'obtenir les meilleurs performances possibles. - **HINT** : spécifiez le hint dont vous avez besoin, à l'aide de la syntaxe `/*+ */`. - **POSITION** : spécifiez où positionner le hint dans une instruction SQL. - **SQL STMT** : sélectionnez l'instruction SQL que vous devez utiliser.
	Convert columns and table to uppercase	Cochez cette case pour passer les noms de colonnes et de tables en majuscule.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Use Batch	Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.
	Batch Size	Spécifiez le nombre d'enregistrements à traiter dans chaque lot. La valeur par défaut de ce champ est *10000*. Ce champ est disponible uniquement lorsque la case **Use Batch** est cochée.
	Support null in "SQL WHERE" statement	Cochez cette case pour prendre en compte les valeurs Null lors d'une instruction WHERE.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Ce composant doit être utilisé en tant que composant de sortie. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Oracle. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé au composant **tOracleOutput**, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput.**
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tOracleOutputBulk

Propriétés du tOracleOutputBulk

Les composants **tOracleOutputBulk** et **tOracleBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tOracleOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/Oracle	
Fonction	Ecrit un fichier composé de colonnes et basé sur le délimiteur défini et sur les standards Oracle.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Oracle.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer. *Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.*
	Create directory if not exists	Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Encoding Type	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Bulk file parameters	Configurez les paramètres **Buffer Size** et **StringBuilder Size** afin d'optimiser les performances selon la mémoire.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tOracleBulkExec**. Ensemble ils offrent un gain de performance important pour l'alimentation d'une base de données Oracle.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour des scénarios associés au composant **tOracleOutputBulk**, consultez les scénarios suivants :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tOracleOutputBulkExec

Propriétés du tOracleOutputBulkExec

Les composants **tOracleOutputBulk** et **tOracleBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tOracleOutputBulkExec**.

Famille de composant	Databases/Oracle	
Fonction	Le **tOracleOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Ce composant est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Oracle.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Liste déroulante des pilotes disponibles : **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle. **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante. **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.
	Host	Adresse IP du serveur de la base de données.

		Seuls *localhost*, *127.0.0.1* ou l'adresse IP exacte de la machine locale permettent un fonctionnement optimal. Le serveur de la base de données doit être installé sur la même machine que le *Studio Talend* ou le Job contenant un **tOracleOutputBulkExec**.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	File Name	Nom du fichier à générer et à charger. *Ce fichier est généré sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer un répertoire où stocker la table de sortie si elle n'existe pas.
	Append	Cochez cette case afin d'ajouter de nouvelles lignes à la fin de l'enregistrement.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Truncate** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations d'Update. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (**Advanced settings**) pour une utilisation optimisée de ces opérations.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

		Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
	Field separator	Caractère, chaîne de caractères ou expression régulière pour séparer les champs.
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Use existing control file	Cochez cette case si vous utilisez un fichier de contrôle (.ctl).
	Record format	Définissez le format de l'enregistrement : **Default** : les paramètres du format sont définis par défaut. **Stream** : déterminez le séparateur de fin d'enregistrement. **Fixed** : déterminez la longueur d'enregistrement. **Variable** : déterminez la taille du champ spécifiant la longueur de l'enregistrement.
	Specify .ctl file's INTO TABLE clause manually	Cochez cette case pour renseigner manuellement la clause INTO TABLE du fichier de contrôle, directement dans le code.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs : **None** : aucun caractère de séparation. **Whitespace** : le caractère de séparation est un espace. **EOF (used for loading LOBs from lobfile)** : la séparation est constituée d'un caractère de fin de fichier (End Of File). **Other terminator** : la séparation est constituée d'un caractère autre que ceux précédemment énoncés.
	Use fields enclosure	Cochez cette case si vous voulez utiliser des caractères pour entourer le texte : **Fields enclosure (left part)** : caractère délimitant les champs sur la gauche. **Field enclosure (right part)** : caractère délimitant les champs sur la droite.
	Use schema's Date Pattern to load Date field	Cochez cette case pour utiliser le modèle de date contenu dans le schéma pour remplir le champ concernant la date.
	Specify field condition	Cochez cette case pour définir une condition de chargement des données.

	Preserve blanks	Cochez cette case pour conserver les espaces vides.
	Trailing null columns	Cochez cette case pour charger des données avec des colonnes vides.
	Load options	Cliquez sur le bouton **[+]** afin d'ajouter des options de chargement des données : **Parameter** : choisissez l'un des paramètres de chargement à partir de la liste déroulante. **Value** : définissez la valeur à attribuer au paramètre.
	NLS Language	A partir de la liste déroulante, définissez la langue utilisée pour les données qui ne sont pas codées en Unicode.
	Set Parameter NLS_TERRITORY	Cochez cette case pour modifier les conventions d'appellation utilisées pour la numérotation des jours et des semaines. La valeur par défaut est celle du système d'exploitation.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Oracle encoding type	Sélectionnez le type d'encodage spécifique à Oracle pour le traitement des données. Ce type d'encodage permet au **tOracleOutputBulkExec** de générer correctement la commande Oracle de masse.
	Output	Sélectionnez dans la liste déroulante l'endroit où rediriger le flux de sortie : **to console** : vers la console de log. **to global variable** : vers une variable globale.
	Convert columns and table names to uppercase	Cochez cette case pour mettre le nom des colonnes et celui de la table en majuscules.
	Bulk file parameters	Configurez les paramètres **Buffer Size** et **StringBuilder Size** afin d'optimiser les performances selon la mémoire.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	
Limitation	Le serveur/client de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tOracleOutputBulkExec**, afin que le composant fonctionne correctement.	

Scénarios associés

Pour des scénarios associés au composant **tOracleOutputBulkExec**, consultez les scénarios suivants :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tOracleRollback

Propriétés du tOracleRollback

Famille de composant	Databases	
Fonction	Le **tOracleRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tOracleConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Oracle, notamment les composants **tOracleConnection** et **tOracleCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tOracleRollback**, consultez *tMysqlRollback*.

tOracleRow

Propriétés du tOracleRow

Famille de composant	Databases/Oracle	
Fonction	Le **tOracleRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tOracleRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Liste déroulante des pilotes disponibles: **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle. **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante. **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.

	Database	Nom de la base de données.
	Schema	Nom du schéma Oracle.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Table name	Nom de la table de base de données.
	Query type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	*Use NB_LINE_*	Cette option vous permet d'alimenter la variable afin de passer au sous-job ou au composant suivant le nombre de lignes mises à jour/supprimées/insérées. Ce champ s'applique uniquement si la requête saisie dans le champ **Query** est une requête INSERT, UPDATE ou DELETE. • **NONE** : n'alimente pas la variable. • **INSERTED** : alimente la variable avec le nombre de lignes insérées. • **UPDATED** : alimente la variable avec le nombre de lignes mises à jour. • **DELETED** : alimente la variable avec le nombre de lignes supprimées.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. *Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l'alias de la source de données spécifié.*

		Cette option est indisponible lorsque la case **Use an existing connection** est cochée.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance. Vous pouvez également utiliser les PreparedStatement afin d'éviter les injections SQL. Pour un scénario utilisant cette fonctionnalité, consultez *Scénario 2 : Utiliser l'instance PreparedStatement pour faire une requête sur des données*.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du **tMysqlRow**.
- *Scénario 2 : Utiliser l'instance PreparedStatement pour faire une requête sur des données* du **tMysqlRow**.

tOracleSCD

Le composant **tOracleSCD** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tOracleSCD*.

tOracleSCDELT

Le composant **tOracleSCDELT** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tOracleSCDELT*.

tOracleSP

Propriétés du tOracleSP

Famille de composant	Databases/Oracle	
Fonction	Le composant **tOracleSP** appelle une procédure stockée de base de données.	
Objectif	Ce composant permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Connection type	Liste déroulante des pilotes disponibles: **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle. **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante. **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.
	Property type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier des propriétés du composant. Les champs suivants sont alors pré-remplis.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.

	Username et *Password*	Informations d'authentification sur l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit schema*	Dans une procédure stockée, le schéma est un paramètre d'entrée. Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le *Guide utilisateur du Studio Talend*.
	SP Name	Saisissez le nom exact de la Procédure Stockée (SP).
	Is Function / Return result in	Cochez cette case, si une seule valeur doit être retournée. Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.
	Parameters	Cliquez sur le bouton **[+]** et sélectionnez dans le champ **Schema Columns** les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure.
		Sélectionnez le **Type** de paramètre : **IN** : paramètre d'entrée (Input). **OUT** : paramètre de sortie (Output)/valeur retournée. **IN OUT** : les paramètres d'entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction). **RECORDSET** : les paramètres d'entrée doivent être retournées sous forme d'ensemble de valeurs, au lieu d'une valeur unique. Consultez *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.
		Le type **Custom Type** est utilisé quand une colonne du schéma (**Schema Column**) que vous souhaitez utiliser est définie par l'utilisateur. Deux types de Custom sont disponibles dans le tableau **Parameters**. Pour le premier type : - Cochez la case quand la colonne du schéma correspondant est définie par l'utilisateur. - Si toutes les colonnes du schéma dans le tableau **Parameters** sont de type personnalisé, vous pouvez cocher la case devant **Custom Type**.

		Sélectionnez un type de base de données dans la liste **DB Type** afin de mapper le type de la base de données source vers le type de la base de données cible : - **Auto-Mapping** : Mappe le type de la base de données source au type de la base de données cible automatiquement (par défaut). - **CLOB** : Character large object. - **BLOB** : Binary large object. - **DECIMAL** : objet décimal numérique. - **NUMERIC** : caractère compris entre 0 et 9.
		Dans la seconde colonne **Custom Type**, vous pouvez préciser le type de personnalisation. Il peut être - **STRUCT** : utilisé pour un élément. - **ARRAY** : utilisé pour plusieurs éléments.
		Dans la colonne **Custom name**, spécifiez le type de personnalisation. *Dans le tableau **Parameters**, si une colonne est de **Type** OUT, assurez-vous que son type dans le schéma est bien Object.*
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	NLS Language	A partir de la liste déroulante, définissez la langue utilisée pour les données qui ne sont pas codées en Unicode.
	NLS Territory	Cochez cette case pour modifier les conventions d'appellation utilisées pour la numérotation des jours et des semaines. La valeur par défaut est celle du système d'exploitation.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d'entrée sont autorisés.	
Limitation	La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données. Quand le paramètre défini dans ce composant est personnalisé (**Custom Type**), les composants de la famille **tJava** doivent être placés avant le **tOracleSP**, afin que les utilisateurs puissent définir des valeurs pour les paramètres personnalisés, ou bien après le **tOracleSP**, pour lire et écrire les paramètres personnalisés de type OUT.	

Scénario : Vérifier le format de numéros à l'aide d'une procédure stockée

Le Job suivant permet de se connecter à une base Oracle contenant des numéros de sécurité social et le nom de leur propriétaire, en appelant une procédure stockée qui vérifie le format des numéros en fonction du format standard ###-##-####. Puis le résultat de la vérification, 1 pour un format valide et 0 pour un format non valide, est affiché en sortie dans la console de la vue **Run**.

- Cliquez et déposez les composants suivants : **tOracleConnection**, **tOracleInput**, **tOracleSP** et **tLogRow**.
- Reliez le composant **tOracleConnection** au **tOracleInput** à l'aide d'un lien **Then Run** puisqu'aucune donnée n'est traitée ici.
- Connectez les autres composants à l'aide de liens **Row Main** puisque des lignes sont transmises comme paramètres au composant SP et à la console.
- Dans l'onglet **Basic settings** du composant **tOracleConnection**, paramétrez les informations de connexion à la base de données correspondante. Vous pourrez donc réutiliser cette information dans tout autre composant lié aux bases de données.
- Puis double-cliquez sur le composant **tOracleInput** et paramétrez ses propriétés.

- Cochez la case **Use an existing connection** et sélectionnez le composant **tOracleConnection** dans la liste pour réutiliser les informations de connexion déjà paramétrées.
- Sélectionnez l'option **Repository** dans le champ **Property type** puisque le schéma Oracle est défini dans une métadonnée du dossier **Db Connections** du **Repository**. Si vous n'avez pas enregistré de métadonnée contenant les informations de connexion à la base de données Oracle dans le **Repository**, renseignez le nom du schéma manuellement.
- Puis sélectionnez l'option **Repository** dans le champ **Schema**, et récupérez le schéma correspondant à la table de la base de données Oracle.

	ID	NAME	CITY	SSNUMBER
1	1	Jack	LA	123-45-6789
2	2	Tom	NYC	123-A5-6789
3	3	Bill	SF	123=45-6789
4	4	Jana	NYC	236-52-2956
5	5	Brandon	SLC	561-52-B267

- Dans cet exemple, la table contenant le numéro de sécurité social contient un schéma à quatre colonnes dont : *ID*, *NAME*, *CITY* et *SSNUMBER*.

- Dans le champ **Query**, saisissez la requête Select suivante ou sélectionnez-la dans la liste si vous l'aviez stockée dans le Repository.

```
select ID, NAME, CITY, SSNUMBER from SSN
```

- Puis sélectionnez le **tOracleSP** et paramétrez ses propriétés dans l'onglet **Basic settings**.

- Comme pour le composant **tOracleInput**, sélectionnez l'option **Repository** dans le champ **Property type** et cochez la case **Use an existing connection**, puis sélectionnez l'entrée adéquate dans la liste **Component List**.

- Le schéma utilisé pour le composant **tOracleSP** diffère légèrement du schéma d'entrée. En effet, une colonne supplémentaire (*SSN_Valid*) est ajoutée au schéma d'entrée. Cette colonne contiendra le statut de validité (*1* ou *0*) généré par la procédure.

- Puis sélectionnez le type d'encodage adéquat dans la liste **Encoding type**.

- Dans le champ **SP Name**, saisissez le nom exact de la procédure stockée (ou de la fonction) telle qu'elle est appelée dans la base de données. Dans cet exemple, le nom de la procédure stockée est *is_ssn*.

- La fonction de base utilisée pour cet exemple est :

```
CREATE OR REPLACE FUNCTION is_ssn(string_in VARCHAR2)
RETURN PLS_INTEGER
IS
-- validating ###-##-#### format
BEGIN
  IF TRANSLATE(string_in, '0123456789A', 'AAAAAAAAAAB') =
     'AAA-AA-AAAA' THEN
    RETURN 1;
  END IF;
  RETURN 0;
END is_ssn;
/
```

- Comme un valeur retournée est attendue dans cet exemple, la procédure agit comme une fonction, ainsi cochez la case **Is function**.

- La seule valeur retournée attendue est basée sur la colonne *ssn_valid*. Ainsi, sélectionnez l'entrée correspondante dans la liste.
- Dans la zone **Parameters**, définissez les paramètres d'entrée et de sortie utilisés dans la procédure. Dans cet exemple, seule la colonne *SSNumber* du schéma est utilisée dans la procédure.
- Cliquez sur le signe **[+]** pour ajouter un nouvelle ligne à la table et sélectionnez la colonne adéquate (*SSNumber*) et son type (*IN*).
- Puis sélectionnez le composant **tLogRow** et cliquez sur **Sync Column** pour récupérer le schéma du composant précédent (**tOracleSP**).

- Cochez la case **Print values in cells of a table** pour faciliter la lecture de la sortie.
- Puis enregistrez le Job et appuyez sur **F6** pour l'exécuter.

```
Starting job OracleSP at 15:14 23/08/2007.
.---+-------+----+-----------+---------.
|              tLogRow_1                |
|=--+-------+----+-----------+--------=|
|ID |NAME   |CITY|SSNUMBER   |SSN_Valid|
|=--+-------+----+-----------+--------=|
|1.0|Jack   |LA  |123-45-6789|1        |
|2.0|Tom    |NYC |123-A5-6789|0        |
|3.0|Bill   |SF  |123=45-6789|0        |
|4.0|Jana   |NYC |236-52-2956|1        |
|5.0|Brandon|SLC |561-52-B267|0        |
'---+-------+----+-----------+---------'

Job OracleSP ended at 15:14 23/08/2007. [exit code=0]
```

Dans la console, vous pouvez lire les résultats en sortie. Toutes les colonnes du schéma d'entrée sont affichées même si elles ne sont pas utilisées comme paramètres dans la procédure stockée.

La colonne finale affiche la valeur retournée attendue, c'est-à-dire que le numéro de sécurité sociale vérifié soit valide ou non.

Consultez également *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario : Récupérer des informations personnelles à l'aide d'une procédure stockée*.
- *Scénario : Trouver le libellé State à l'aide d'une procédure stockée*.
- *Scénario : Exécuter une procédure stockée dans le hub MDM*.

tOracleTableList

Propriétés du tOracleTableList

Famille de composant	Databases/Oracle	
Fonction	Le composant **tOracleTableList** effectue une opération d'itération sur toutes les tables d'une base de données, grâce à une connexion Oracle définie.	
Objectif	Le composant **tOracleTableList** permet de lister les noms des tables Oracle grâce aux commandes SELECT et WHERE.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tOracleConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Where clause for table name selection	Saisissez la commande WHERE permettant d'identifier les tables sur lesquelles effectuer l'opération d'itération.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**CURRENT_TABLE** : nom de la table sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_TABLE** : nombre de tables itérées jusqu'à présent. Cette variable est une variable **Flow** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Il faut utiliser ce composant en association avec les autres composants Oracle, notamment avec le **tOracleConnection**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Itérer une table de base de données et lister le nom des colonnes de la table*.

tPostgresqlBulkExec

Propriétés du tPostgresqlBulkExec

Les composants **tPostgresqlOutputBulk** et **tPostgresqlBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tPostgresqlOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/ Postgresql	
Fonction	Le composant **tPostgresqlBulkExec** permet d'insérer les données d'une base de données Postgresql.	
Objectif	Le **tPostgresqlBulkExec** est un composant dédié, il permet un gain de performance lors du traitement des données de la base de données Postgresql.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle.

		Create table : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	File Name	Nom du fichier à charger. *Le fichier est situé sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
Advanced settings	*Action on data*	Sélectionnez l'opération que vous voulez effectuer : **Bulk insert** : ajoute de nombreuses entrées dans la table. **Bulk update** : effectue des modifications simultanées dans différentes entrées.
	Copy the OID for each row	Récupère les identifiants d'objet pour chaque ligne.
	Contains a header line with the names of each column in the file	Spécifiez que la table contient des lignes d'en-tête.
	File type	Sélectionnez le type de fichiers à traiter.
	Null string	Chaîne de caractères affichée pour signifier que la valeur est nulle.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs.
	Escape char	Caractère d'échappement de la ligne.
	Text enclosure	Caractères utilisés pour entourer le texte.
	Use standard_conforming_string ON	Activez la variable.

	Force not null for columns	Définissez la nullabilité des colonnes. **Force not null** : Cochez la case correspondant à la colonne (**Column**) que vous souhaitez définir comme n'étant pas nulle.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être utilisé en association avec le composant **tPostgresqlOutputBulk**. Ensemble, ils permettent d'obtenir des gains de performances dans une base de données Postgresql.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour d'autres exemples d'utilisation du composant **tPostgresqlBulkExec**, consultez les scénarios suivants dans :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**,
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**,
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tPostgresqlCommit

Propriétés du tPostgresqlCommit

Famille de composant	Databases/Postgresql	
Fonction	Le composant **tPostgresqlCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.	
Basic settings	*Component list*	Sélectionnez le composant **tPostgresqlConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row>Main** pour relier le **tPostgresqlCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Postgresql, notamment les composants **tPostgresqlConnection** et **tPostgresqlRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tPostgresqlCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tPostgresqlClose

Propriétés du tPostgresqlClose

Famille de composant	Databases/Postgresql	
Fonction	Le composant **tPostgresqlClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Postgresql.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tPostgresqlConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tPostgresqlConnection** et **tPostgresqlRollback**. Il est généralement utilisé avec un composant **tPostgresqlConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tPostgresqlClose**, consultez *tMysqlConnection*.

tPostgresqlConnection

Propriétés du tPostgresqlConnection

Famille de composant	Databases/Postgresql	
Fonction	Le composant **tPostgresqlConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

		Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants Postgresql, notamment les composants **tPostgresqlCommit** et **tPostgresqlRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tPostgresqlConnection**, consultez *tMysqlConnection*.

tPostgresqlInput

Propriétés du tPostgresqlInput

Famille de composant	Databases/ Postgresql	
Fonction	Le composant **tPostgresqlInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tPostgresqlInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.

		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Table Name	Nom de la table à lire.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Use cursor*	Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d'optimiser les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Postgresql.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez les scénarios du composant **tPostgresqlInput** :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

tPostgresqlOutput

Propriétés du tPostgresqlOutput

Famille de composant	Databases/ Postgresql	
Fonction	Le composant **tPostgresqlOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	**Le tPostgresqlOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore.

		Create table if not exists : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
	Use spatial options	Cette case est disponible lorsque vous sélectionnez l'option de création de table dans la liste **Action on table** et disparaît si vous cochez la case **Enable parallel execution** dans la vue **Advanced settings**.

		Cochez cette case pour utiliser la base de données PostgreSQL en tant que base de données spatiale pour un système d'informations géographique (SIG). Lorsque cette case est cochée, deux autres cases s'affichent : • **Create Spatial index**: cochez cette case pour créer un index spatial. • **Create geometry columns reference** : cochez cette case pour créer la référence des colonnes de type Geometry. Pour plus d'informations concernant la base de données PostgreSQL, consultez http://www.postgresql.org/about/.
	Die on error	Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tPostgresqlOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use save point	Cochez cette case pour utiliser des points de sauvegarde au cours de la transaction. Cette case est indisponible si vous cochez : • la case **Die on error** dans la vue **Basic settings**, ou • la case **Use Batch Size** dans la vue **Advanced settings**. Cette case ne fonctionne pas si vous : • saisissez *0* dans le champ **Commit every**, ou • cochez la case **Use an existing connection** dans la vue **Basic settings** et que le mode **Auto Commit** est activé dans le composant de connexion à la base de données.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Support null in "SQL WHERE" statement	Cochez cette case pour prendre en compte les valeurs Null d'une table de base de données. Assurez-vous que la case **Nullable** est bien cochée pour les colonnes du schéma correspondantes.
	Use Batch Size	Cochez cette case pour activer le mode Batch (par lots) pour le traitement des données. Dans le champ **Batch Size** qui apparaît lorsque la case est cochée, saisissez la taille du lot à traiter.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**.	

	Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Postgresql. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé au composant **tPostgresqlOutput** consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du **tMysqlOutput**.

tPostgresqlOutputBulk

Propriétés du tPostgresqlOutputBulk

Les composants **tPostgresqlOutputBulk** et **tPostgresqlBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tPostgresqlOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/Postgresql	
Fonction	Ecrit un fichier composé de colonnes et basé sur le délimiteur défini et sur les standards Postgresql.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Postgresql.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer. *Le fichier est généré sur la machine locale ou dans un fichier partagé sur le réseau local.*
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
Advanced settings	*Row separator*	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tPostgresqlBulkExec**. Ensemble ils offrent un gain de performance important pour l'alimentation d'une base de données Postgresql.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé au composant **tPostgresqlOutputBulk**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tPostgresqlOutputBulkExec

Propriétés du tPostgresqlOutputBulkExec

Les composants **tPostgresqlOutputBulk** et **tPostgresqlBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tPostgresqlOutputBulkExec**.

Famille de composant	Databases/Postgresql	
Fonction	Le **tPostgresqlOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tPostgresqlOutputBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Postgresql.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données. *localhost*, *127.0.0.1* et l'adresse IP exacte de la machine locale permettent un bon fonctionnement. Le serveur de base de données doit être installé sur la même machine que le *Studio Talend* ou le Job comprenant le **tPostgresqlOutputBulkExec**.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None**: n'effectuer aucune opération de table. **Drop and create table**: supprimer la table puis en créer une nouvelle. **Create table**: créer une table qui n'existe pas encore. **Create table if not exists**: créer la table si nécessaire. **Drop table if exists and create**: supprimer la table si elle existe déjà, puis en créer une nouvelle.

		Clear a table: supprimer le contenu de la table.
	File Name	Nom du fichier à générer et à charger. *Le fichier est généré sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
Advanced settings	*Action on data*	Sélectionnez l'opération que vous voulez effectuer : **Bulk insert** : ajoute différentes entrées à la table. Si des doublons sont trouvés, le Job s'arrête. **Bulk update** : effectue des modifications simultanées dans différentes entrées.
	Copy the OID for each row	Récupère les identifiants d'objet pour chaque ligne.
	Contains a header line with the names of each column in the file	Spécifiez que la table contient des lignes d'en-tête.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	File type	Sélectionnez le type de fichiers à traiter.
	Null string	Chaîne de caractères affichée pour signifier que la valeur est nulle.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs.
	Escape char	Caractère d'échappement de la ligne.
	Text enclosure	Caractères utilisés pour entourer le texte.

	Activate standard_conforming_string	Activez le paramètre standard_conforming_string pour traiter les backslashs comme une chaîne ordinaire et non pas comme un caractère d'échappement.
	Force not null for columns	Définissez la nullabilité des colonnes. **Force not null** : Cochez la case correspondant à la colonne (**Column**) que vous souhaitez définir comme n'étant pas nulle.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	
Limitation	Le serveur de base de données doit être installé sur la même machine que le *Studio Talend* ou le Job comprenant un **tPostgresqlOutputBulkExec** afin que le composant fonctionne correctement.	

Scénarios associés

Pour un scénario associé au composant **tPostgresqlOutputBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tPostgresqlRollback

Propriétés du tPostgresqlRollback

Famille de composant	Databases/Postgresql	
Fonction	Le **tPostgresqlRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tPostgresqlConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Postgresql, notamment les composants **tPostgresqlConnection** et **tPostgresqlCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tPostgresqlRollback**, consultez *tMysqlRollback*.

tPostgresqlRow

Propriétés du tPostgresqlRow

Famille de composant	Databases/ Postgresql	
Fonction	Le **tPostgresqlRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tPostgresqlRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Query type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tPostgresqlSCD

Le composant **tPostgresqlSCD** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tPostgresqlSCD*.

tPostgresqlSCDELT

Le composant **tPostgresqlSCDELT** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tPostgresqlSCDELT*.

tSybaseBulkExec

Propriétés du tSybaseBulkExec

Les composants **tSybaseOutputBulk** et **tSybaseBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tSybaseOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases	
Fonction	Le composant **tSybaseBulkExec** permet d'insérer les données d'une base de données Sybase.	
Objectif	Le **tSybaseBulkExec** est un composant dédié, il permet un gain de performance lors du traitement des données de la base de données Sybase.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Server	Adresse IP du serveur de base de données. Cette option est disponible lorsque vous sélectionnez **Bulk Update** dans la liste **Action on data**, dans la vue **Advanced settings** du **tSybaseBulkExec**.
	Port	Numéro du port d'écoute du serveur de base de données. Cette option est disponible lorsque vous sélectionnez **Bulk Update** dans la liste **Action on data**, dans la vue **Advanced settings** du **tSybaseBulkExec**.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Sybase Schema	Nom du schéma de la base de données que vous souhaitez utiliser.
	Server	Adresse IP du serveur de base de données pour la connexion de l'utilitaire Bcp.
	Bcp Utility	Nom de l'utilitaire bcp à utiliser pour copier les données sur le serveur Sybase.

	Batch size	Nombre de lignes dans chaque lot de traitement de données.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération sur la table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	File Name	Nom du fichier à charger. *Ce fichier doit se situer sur la même machine que le serveur de la base de données.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Use an interface file*	Cochez cette case afin de spécifier un fichier interface dans le champ **Interface file**.
	Additional JDBC parameters	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple : CHARSET=KANJISIS_OS pour supporter les caractères japonais.
	Action on data	Vous pouvez effectuer les actions suivantes sur la table définie : **Bulk Insert** : ajoute des entrées multiples à la table. Le Job s'arrête lorsque des doublons sont trouvés. **Bulk Update** : effectue des changements simultanés aux multiples entrées.
	Field Terminator	Caractère, chaîne ou expression régulière séparant les champs.
	Row Terminator	Chaîne (ex : "\n" sous Unix) séparant les lignes.

	Head row	Nombre de lignes d'en-tête à ignorer en début de fichier.
	Encoding	Sélectionnez l'encodage dans la liste ou saisissez-le entre guillemets doubles si le type d'encodage utilisé n'existe pas dans la liste. Ce champ est obligatoire pour la gestion de données de bases de données.
	Output	Sélectionnez la sortie à utiliser pour récupérer la sortie standard de la base Sybase : **to console** : vers la console. **to global variable** : vers la variable globale.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job , ainsi qu'au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE_DATA** : nombre de lignes lues. Cette variable est une variable **After** et retourne un entier. **NB_LINE_BAD** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé lorsque les données à charger dans la base n'exigent aucune transformation particulière.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Le serveur/client de la base de données doit être installé sur la même machine que le *Studio Talend* ou le Job contenant un **tSybaseBulkExec**, afin que le composant fonctionne correctement. Contrairement au composant Oracle dédié, ce composant Sybase ne permet aucune action directe sur les données. Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé au composant **tSybaseBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tSybaseClose

Propriétés du tSybaseClose

Famille de composant	Databases/Sybase	
Fonction	Le composant **tSybaseClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Sybase.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tSybaseConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tSybaseConnection** et **tSybaseRollback**. Il est généralement utilisé avec un composant **tSybaseConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tSybaseClose**, consultez *tMysqlConnection*.

tSybaseCommit

Propriétés du tSybaseCommit

Famille de composant	Databases/Sybase	
Fonction	Le composant **tSybaseCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	Sélectionnez le composant **tSybaseCommit** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close Connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tSybaseCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Sybase, notamment les composants **tSybaseConnection** et **tSybaseRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tSybaseCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tSybaseConnection

Propriétés du tSybaseConnection

Famille de composant	Databases/Sybase	
Fonction	Le composant **tSybaseConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

		Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants Sybase, notamment les composants **tSybaseCommit** et **tSybaseRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé au composant **tSybaseConnection**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tSybaseInput

Propriétés du tSybaseInput

Famille de composant	Databases/ Sybase	
Fonction	Le composant **tSybaseInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tSybaseInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Server	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Sybase Schema	Nom exact du schéma Sybase.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à lire.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Trim all the String/Char columns*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Sybase.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez les scénarios du composant **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du **tContextLoad**.

tSybaseIQBulkExec

Propriétés du tSybaseIQBulkExec

Famille de composant	Databases/SybaseIQ	
Fonction	Le composant **tSybaseIQBulkExec** permet de charger un fichier bulk dans une base de données Sybase IQ.	
Objectif	Le **tSybaseIQBulkExec** est un composant dédié, il permet un gain de performance lors du traitement des données de la base de données Sybase IQ.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Les versions disponibles de Sybase sont : - **Sybase IQ 12**, - **Sybase IQ15**. • *Pour **Sybase IQ 12**, le fichier de masse à charger doit se situer sur la même machine que le serveur de **Sybase IQ 12**.* • *Pour **Sybase IQ 15**, le fichier de masse à charger doit se trouver dans une machine distante. Cependant, cela signifie de configurer le serveur **Sybase IQ 15**. Pour plus d'informations, consultez Sybase IQ client-side load support enhancements (en anglais).* Vous pouvez vous connecter à la version **Sybase IQ 15** via ODBC et via JDBC pour la version **Sybase IQ 12**. Les champs à renseigner dans la vue **Basic settings** varient légèrement selon les versions.
	Use an existing connection *Sybase IQ 12 uniquement.*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host *Sybase IQ 12 uniquement.*	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.

	Sybase IQ 12 uniquement.	
	Data Source *Sybase IQ 15 uniquement.*	Sélectionnez le type de source de données à utiliser et renseignez les informations de DSN dans les champs correspondants. Les types disponibles sont : - **DSN**, - **FILEDSN**. Lorsque le type **FILEDSN** est utilisé, un bouton **[...]** apparaît près du champ **Data Source** et vous permet de parcourir le fichier source de données qui vous intéresse.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération sur la table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Local filename	Nom du fichier à charger.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit **Built-in** soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*Lines terminated by*	Caractère, chaîne ou expression régulière séparant les lignes.
	Field Terminated by	Caractère, chaîne ou expression régulière séparant les champs. Ce composant permet l'utilisation de séparateurs de champs/lignes orientés Sybase, comme \x09.
	Use enclosed quotes	Cochez cette case pour utiliser des caractères pour entourer les données.

	Use fixed length	Cochez cette case pour déterminer une longueur fixe pour les lignes de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données Sybase IQ.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Contrairement au composant Oracle dédié, ce composant Sybase ne permet aucune action directe sur les données. Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*. Le fichier jodbc.jar doit être installé séparément dans la vue **Modules** de la perspective **Integration** de votre *Studio Talend*.Pour plus d'informations, consultez la section concernant les modules externes dans le *Guide d'installation et de migration Talend*. *Pour **Sybase IQ 12**, le serveur/client de la base de données doit être installé sur la même machine que le Studio Talend ou que le Job contenant un **tSybaseIQOutputBulkExec**.* *Pour **Sybase IQ 15**, vous pouvez installer uniquement le client de la base de données sur la même machine que le Studio Talend ou que le Job contenant un **tSybaseIQOutputBulkExec**. Cependant, cela nécessite une certaine configuration du serveur de **Sybase IQ 15**. Pour plus d'informations, consultez Sybase IQ client-side load support enhancements (en anglais).*	

Scénario associé

Pour un scénario associé au **tSybaseIQBulkExec**, consultez :

- *Scénario : Charger en masse de données dans une base de données Sybase IQ 12* du **tSybaseIQOutputBulkExec**.
- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tSybaseIQOutputBulkExec

Propriétés du tSybaseIQOutputBulkExec

Famille de composant	Databases/SybaseIQ	
Fonction	Le **tSybaseIQOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tSybaseIQOutputBulkExec** est un composant dédié, il permet un gain de performance lors du traitement des données de la base de données Sybase IQ.	
Basic settings	*Property type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Les versions disponibles de Sybase sont : - **Sybase IQ 12**, - **Sybase IQ15**. • *Pour **Sybase IQ 12**, le fichier de masse à charger doit se situer sur la même machine que le serveur de **Sybase IQ 12**.* • *Pour **Sybase IQ 15**, le fichier de masse à charger doit se trouver dans une machine distante. Cependant, cela signifie de configurer le serveur **Sybase IQ 15**. Pour plus d'informations, consultez Sybase IQ client-side load support enhancements (en anglais).* Vous pouvez vous connecter à la version **Sybase IQ 15** via ODBC et via JDBC pour la version **Sybase IQ 12**. Les champs à renseigner dans la vue **Basic settings** varient légèrement selon les versions.
	Use an existing connection *Sybase IQ 12 uniquement.*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host *Sybase IQ 12 uniquement.*	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.

	Sybase IQ 12 uniquement.	
	Data Source *Sybase IQ 15 uniquement.*	Sélectionnez le type de source de données à utiliser et renseignez les informations de DSN dans les champs correspondants. Les types disponibles sont : - **DSN**, - **FILEDSN**. Lorsque le type **FILEDSN** est utilisé, un bouton **[...]** apparaît près du champ **Data Source** et vous permet de parcourir le fichier source de données qui vous intéresse.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération sur la table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Create table if doesn't exist** : créer la table si nécessaire. **Clear a table** : supprimer le contenu de la table.
	File Name	Nom du fichier à générer et à charger.
	Append the file	Cochez cette case pour ajouter des lignes à la fin des enregistrements.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job

		courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs. *Ce composant étant une combinaison des **tSybaseOutputBulk** et **tSybaseIQBulkExec**, il ne permet pas l'utilisation de séparateurs de champs/lignes orientés Sybase, tels que \x09. Pour obtenir l'effet souhaité, (par exemple afficher les champs sous forme de tableau) vous devez utiliser le **tSybaseOutputBulk** et le **tSybaseIQBulkExec** ensemble afin de remplacer le **tSybaseOutputBulkExec**, avec le séparateur \t utilisé dans le **tSybaseOutputBulk** et le séparateur \x09 utilisé dans le **tSybaseIQBulkExec**.*
	Lines terminated by	Caractère, chaîne ou expression régulière séparant les lignes.
	Use enclose quotes	Cochez cette case pour utiliser des caractères pour entourer les données.
	Include Head	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*. Le fichier jodbc.jar doit être installé séparément dans la vue **Modules** de la perspective **Integration** de votre *Studio Talend*.Pour plus d'informations, consultez la section concernant les modules externes dans le *Guide d'installation et de migration Talend*. *Pour **Sybase IQ 12**, le serveur/client de la base de données doit être installé sur la même machine que le Studio Talend ou que le Job contenant un **tSybaseIQOutputBulkExec**.* *Pour **Sybase IQ 15**, vous pouvez installer uniquement le client de la base de données sur la même machine que le Studio Talend ou que le Job contenant un **tSybaseIQOutputBulkExec**. Cependant, cela nécessite une certaine configuration du serveur de **Sybase IQ 15**. Pour plus d'informations, consultez Sybase IQ client-side load support enhancements (en anglais).*	

Scénario : Charger en masse de données dans une base de données Sybase IQ 12

Ce scénario sauvegarde des données provenant d'un **tRowGenerator** dans un fichier et les charge en masse dans une base de données Sybase IQ 12.

Relier les composants

1. Déposer un composant **tRowGenerator** et un **tSybaseIQOutputBulkExec** dans l'espace de modélisation graphique.

2. Reliez le **tRowGenerator** au **tSybaseIQOutputBulkExec** à l'aide d'un lien a **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tRowGenerator** pour ouvrir l'éditeur de schéma.

2. Cliquez sur le bouton **[+]** pour ajouter deux colonnes et nommez-les *id* et *name*.

3. Sélectionnez le type *int* pour la colonne *id* et *String* pour la colonne *name*.

4. Définissez la longueur pour les colonnes *id* et *name*, respectivement *4* et *30*.

5. Sélectionnez la fonction pour *id* et *name*, respectivement `Numeric.sequence` et `TalendDataGenerator.getFirstName`.

6. Cliquez sur **OK** pour fermer l'éditeur et cliquez sur **Yes** sur propager les modifications :

7. Double-cliquez sur le **tSybaseIQOutputBulkExec** pour ouvrir sa vue **Basic settings**.

8. Dans les champs **Host** et **Port**, saisissez les informations de connexion.

9. Dans le champ **Database**, saisissez le nom de la base de données.

10. Dans les champs **Username** et **Password**, saisissez les informations d'authentification.

11. Dans le champ **Table**, saisissez le nom de la table.

12. Dans la liste **Action on table**, sélectionnez **Create table if not exists**.

13. Dans le champ **Filename**, saisissez le chemin d'accès complet au fichier contenant les données.

Exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder le Job.

2. Appuyez sur la touche **F6** pour exécuter le Job.

```
Starting job usecase_sybase at 14:50 14/11/2012.

[statistics] connecting to socket on port 3643
[statistics] connected
[statistics] disconnected
Job usecase_sybase ended at 14:50 14/11/2012. [exit code=0]
```

3. Dans la console **Sybase Central**, ouvrez la table *staff* afin de vérifier les données :

Comme montré ci-dessus, la table est créée et les données sont insérées.

Scénarios associés

Pour un scénario associé au composant **tSybaseIQOutputBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tSybaseOutput

tSybaseOutput Properties

Famille de composant	Databases/ Sybase	
Fonction	Le composant **tSybaseOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	**Le tSybaseOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Server	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Sybase Schema	Nom exact du schéma Sybase.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **Default** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle.

		Create table : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Clear table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Turn on identity insert	Cochez cette case pour utiliser votre propre séquence sur les valeurs Identity des données insérées (plutôt que de laisser le serveur SQL choisir les valeurs séquentielles).
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tSybaseOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Use Batch size	Cochez cette case afin d'activer le mode batch pour le traitement des données. Dans le champ **Batch Size** qui apparaît, vous pouvez saisir le nombre de lignes par lot à charger. Cette case est disponible uniquement si vous avez choisi l'option **Insert**, **Update**, **Single Insert Query** ou **Delete** dans le champ **Action on data**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Sybase. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé au composant **tSybaseOutput**, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du **tMysqlOutput**.

tSybaseOutputBulk

Propriétés du tSybaseOutputBulk

Les composants **tSybaseOutputBulk** et **tSybaseBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tSybaseOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/Sybase	
Fonction	Ecrit un fichier composé de colonnes et basé sur le délimiteur défini et sur les standards Sybase.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Sybase.	
Basic settings	*Property type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer. *Ce fichier est généré sur la machine locale ou dans un dossier partagé du réseau local.*
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Row separator*	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs. *Cette option respecte la syntaxe Java et ne permet pas l'utilisation de séparateurs de lignes/champs orientés Sybase, comme \x09.*
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tSybaseBulkExec**. Ensemble ils offrent un gain de performance important pour l'alimentation d'une base de données Sybase.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au composant **tSybaseOutputBulk**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tSybaseOutputBulkExec

Propriétés du tSybaseOutputBulkExec

Les composants **tSybaseOutputBulk** et **tSybaseBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tSybaseOutputBulkExec**.

Famille de composant	Databases/Sybase	
Fonction	Le **tSybaseOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tSybaseOutputBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Sybase.	
Basic settings	*Property type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Server	Adresse IP du serveur de base de données. Seuls *localhost*, *127.0.0.1* ou l'adresse IP exacte de la machine locale permettent un fonctionnement optimal. Le serveur de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tSybaseOutputBulkExec**.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Bcp Utility	Nom de l'utilitaire bcp à utiliser pour copier les données sur le serveur Sybase.
	Batch row number	Nombre de lignes à traiter dans chaque lot.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table.

		Drop and create the table : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Clear a table** : supprimer le contenu de la table.
	File Name	Nom du fichier à générer et à charger. *Ce fichier est généré sur la machine spécifiée par l'URI dans le champ **Server** et doit être sur la même machine que le serveur de la base de données.*
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Use an interface file*	Cochez cette case afin de spécifier un fichier interface dans le champ **Interface file**.
	Additional JDBC parameters	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple : CHARSET=KANJISIS_OS pour supporter les caractères japonais.
	Action on data	Vous pouvez effectuer les actions suivantes sur la table définie : **Bulk Insert** : ajoute des entrées multiples à la table. Le Job s'arrête lorsque des doublons sont trouvés. **Bulk Update** : effectue des changements simultanés aux multiples entrées.
	Field Terminator	Caractère, chaîne ou expression régulière séparant les champs. *Ce composant étant une combinaison des **tSybaseOutputBulk** et **tSybaseBulkExec**, il ne permet pas l'utilisation de séparateurs de champs/lignes orientés Sybase, tels que \x09. Pour obtenir l'effet souhaité, (par exemple afficher les champs sous forme de tableau) vous devez utiliser le **tSybaseOutputBulk** et le **tSybaseBulkExec** ensemble afin de remplacer le **tSybaseOutputBulkExec**, avec le séparateur \t utilisé dans le **tSybaseOutputBulk** et le séparateur \x09 utilisé dans le **tSybaseBulkExec**.*
	DB Row Terminator	Chaîne (ex : "\n" sous Unix) séparant les lignes dans la base de données.
	First row NO. of file	Nombre de lignes d'en-tête à ignorer en début de fichier.

	FILE Row Terminator	Caractère, chaîne ou expression régulière séparant les lignes dans le fichier.
	Include Head	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Sybase encoding type	Sélectionnez le type d'encodage spécifique à Sybase pour le traitement des données. Ce type d'encodage permet au **tSybaseOutputBulkExec** de générer correctement la commande Sybase de masse.
	Output	**to console** : l'information est chargée. **to global variable** : valeurs retournées à partir de fichiers de log.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	
Limitation	Le serveur/client de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tSybaseOutputBulkExec**, afin que le composant fonctionne correctement.	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé au composant **tSybaseOutputBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tSybaseRollback

Propriétés du tSybaseRollback

Famille de composant	Databases/Sybase	
Fonction	Le **tSybaseRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tSybaseConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Sybase, notamment les composants **tSybaseConnection** et **tSybaseCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tSybaseRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille*.

tSybaseRow

Propriétés du tSybaseRow

Famille de composant	Databases/ Sybase	
Fonction	Le **tSybaseRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tSybaseRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Server	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Sybase schema	Nom du schéma Sybase.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table name	Nom de la table à lire.
	Turn on identity insert	Cochez cette case pour utiliser votre propre séquence sur les valeurs Identity des données insérées (plutôt que de laisser le serveur SQL choisir les valeurs séquentielles).
	Query type	Peut être Built-in ou Repository.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.

	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**.	

	Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé au composant **tSybaseRow**, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du **tMysqlRow**.

tSybaseSCD

Le composant tSybaseSCD appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tSybaseSCD*.

tSybaseSCDELT

Le composant **tSybaseSCDELT** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tSybaseSCDELT*.

tSybaseSP

Propriétés du tSybaseSP

Famille de composant	Databases/ Sybase	
Fonction	Le composant **tSybaseSP** appelle une procédure stockée de base de données.	
Objectif	Ce composant permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier des propriétés du composant. Les champs suivants sont alors pré-remplis.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	SP Name	Saisissez le nom exact de la Procédure Stockée (SP).
	Is Function / Return result in	Cochez cette case, si une seule valeur doit être retournée. Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.
	Timeout Interval	Durée maximale d'attente des résultats de la procédure stockée.
	Parameters	Cliquez sur le bouton **[+]** et sélectionnez dans le champ **Schema Columns** les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure. Sélectionnez le **Type** de paramètre : **IN** : paramètre d'entrée (Input) **OUT** : paramètre de sortie (Output)/valeur retournée **IN OUT** : les paramètres d'entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction). **RECORDSET** : les paramètres d'entrée doivent être retournées sous forme d'ensemble de valeurs, au lieu d'une valeur unique. Consultez *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Use Multiple SELECT Procedure	Cochez cette case pour utiliser des procédures contenant plusieurs instructions SELECT.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d'entrée sont autorisés.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données. Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez:

- *Scénario : Récupérer des informations personnelles à l'aide d'une procédure stockée*.
- *Scénario : Trouver le libellé State à l'aide d'une procédure stockée*.
- *Scénario : Vérifier le format de numéros à l'aide d'une procédure stockée*.
- *Scénario : Exécuter une procédure stockée dans le hub MDM*.

Consultez également *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.

tVerticaSCD

Le composant **tVerticaSCD** appartient à deux familles de composants : Business Intelligence et Databases. Pour plus d'informations, consultez *tVerticaSCD*.

Databases - Composants d'appliance et de datawarehouse

Le chapitre *Databases - Composants d'appliance et de datawarehouse* décrit les connecteurs de bases de données spécialisées dans le traitement de gros volumes de données. Ces connecteurs couvrent divers besoins : ouverture de connexion, lecture et écriture de tables, commit de transactions, ainsi que des rollbacks pour le traitement des erreurs. Ces composants sont présents dans la famille **Databases** de la **Palette** dans la perspective **Integration** du *Studio Talend*. D'autres types de connecteurs de bases de données, tels que les connecteurs des bases de données traditionnelles et de gestion de bases de données, sont documentés dans le *Databases - Composants traditionnels* et le *Databases - Autres composants*.

tGreenplumBulkExec

Propriétés du tGreenplumBulkExec

Les composants **tGreenplumOutputBulk** et **tGreenplumBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tGreenplumOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Famille de composant	Databases/ Greenplum	
Fonction	Le composant **tGreenplumBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tGreenplumBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Greenplum.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore.

		Create table if doesn't exist : créer la table si nécessaire. **Clear a table** : supprimer le contenu de la table.
	File Name	Nom du fichier à charger. *Le fichier est situé sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Action on data*	Sélectionnez l'opération que vous voulez effectuer : **Bulk insert** ou **Bulk update**. Les informations demandées seront différentes en fonction de l'action choisie.
	Copy the OID for each row	Récupère les identifiants d'objet pour chaque ligne.
	Contains a header line with the names of each column in the file	Spécifiez que la table contient des lignes d'en-tête.
	File type	Sélectionnez le type de fichiers à traiter.
	Null string	Chaîne de caractères affichée pour signifier que la valeur est nulle.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs.
	Escaped char	Caractère d'échappement de la ligne.
	Text enclosure	Caractères utilisés pour entourer le texte.
	Force not null for columns	Définissez la nullabilité des colonnes. **Force not null** : Cochez la case correspondant à la colonne (**Column**) que vous souhaitez définir comme n'étant pas nulle.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement	

	dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé avec un composant **tGreenplumOutputBulk**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données Greenplum.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour plus d'informations relatives au fonctionnement de **tGreenplumBulkExec**, consultez les scénarios suivants, dans :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du composant **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du composant **tOracleBulkExec**.

tGreenplumClose

Propriétés du tGreenplumClose

Famille de composant	Databases/Greenplum	
Fonction	Le composant **tGreenplumClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Greenplum.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tGreenplumConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tGreenplumConnection** et **tGreenplumRollback**. Il est généralement utilisé avec un composant **tGreenplumConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tGreenplumClose**, consultez *tMysqlConnection*

tGreenplumCommit

Propriétés du tGreenplumCommit

Famille de composant	Databases/Greenplum	
Fonction	Le composant **tGreenplumCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tGreenplumConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tGreenplumCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Greenplum et notamment avec les composants **tGreenplumConnection** et **tGreenplumRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour consulter des scénarios associés au composant **tGreenplumCommit**, consultez :

- *Scénario: Mapper les données à l'aide d'une jointure implicite simple* .

- *Scénario 1 : Insérer des données dans des tables mère/fille*.

tGreenplumConnection

Propriétés du tGreenplumConnection

Famille de composant	Databases/Greenplum	
Fonction	Le composant **tGreenplumConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

		Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants Greenplum, notamment les composants **tGreenplumCommit** et **tGreenplumRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé au composant **tGreenplumConnection**, consultez :

- *Scénario: Mapper les données à l'aide d'une jointure implicite simple* .
- *tMysqlConnection*.

tGreenplumGPLoad

Ce composant invoque l'utilitaire gpload de Greenplum afin d'insérer des enregistrements dans une base de données Greenplum. Ce composant peut être utilisé soit en mode standalone, il chargera alors les données à partir d'un fichier de données existant, soit connecté à un flux d'entrée, afin de charger des données à partir du composant connecté.

Propriétés du tGreenplumGPLoad

Famille de composant	Databases/Greenplum	
Fonction	Le composant **tGreenplumGPLoad** insère des données dans la table d'une base de données Greenplum via l'utilitaire gpload de Greenplum.	
Objectif	Ce composant permet de charger en masse des données dans une table Greenplum soit à partir d'un fichier, soit à partir d'un flux d'entrée, soit à partir d'un tube nommé.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table dans laquelle les données doivent être insérées.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée, avant de charger les données : **None** : n'effectuer aucune opération de table. **Clear table** : supprimer le contenu de la table. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Drop table if exists and create** : supprimer la table si elle existe déjà et la (re)crée. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Action on data	Vous pouvez effectuer différentes actions sur les données de la table définie :

		Insert : Ajouter de nouvelles entrées à la table. Si des doublons sont trouvés, le Job s'arrête. **Update** : Apporter des modifications aux entrées existantes. **Merge** : Mettre à jour ou ajouter des données dans la table. *Il est nécessaire de spécifier au moins une colonne comme étant une clé primaire sur laquelle baser les opérations **Update** et **Merge**. Vous pouvez le faire en cliquant sur **Edit Schema** et en cochant la (les) case(s) à côté de la (des) colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour configurer les options des opérations **Update** et **Merge**, cochez les cases de la colonne **Match Column** correspondant aux noms des colonnes que vous souhaitez utiliser comme base pour les opérations **Update** et **Merge**. Ensuite, cochez les cases de la colonne **Update Column** correspondant aux noms des colonnes que vous souhaitez mettre à jour. Pour configurer le champ **Update condition**, saisissez la condition qui sera utilisée pour mettre à jour les données.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Data file	Chemin d'accès complet au fichier de données à utiliser. Si ce composant est utilisé seul (non connecté à un autre composant par un flux d'entrée), alors saisissez le nom d'un fichier existant à charger dans la base de données. S'il est connecté à un autre composant par un flux d'entrée, saisissez le nom du fichier à générer et à écrire avec les données d'entrée, afin de l'utiliser plus tard avec gpload pour charger dans la base de données. Ce champ est masqué lorsque la case **Use named-pipe** est cochée.
	Use named-pipe	Cochez cette case afin d'utiliser un tube nommé à la place d'un fichier de données. Cette option ne peut être utilisée que lorsque le composant est connecté à un autre composant par un flux d'entrée. Quand la case est cochée, aucun fichier de données n'est généré, et les données sont transférées à l'utilitaire gpload via un tube nommé. Cette option améliore grandement les performances sous Linux et Windows. Ce composant, en mode tube nommé, utilise une interface JNI pour créer un tube nommé et écrire dedans, dans toute plateforme Windows. Le chemin d'accès associé au fichier

		DLL JNI doit être configuré dans le chemin d'accès à la bibliothèque Java. Le composant est automatiquement fourni dans le *Studio Talend* avec deux DLL pour les systèmes d'exploitation de 32 et 64 bits.
	Named-pipe name	Spécifiez un nom pour le tube nommé à utiliser. Vérifiez que le nom saisi est valide.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Use existing control file (YAML formatted)*	Cochez cette case pour fournir un fichier de contrôle à utiliser avec l'utilitaire gpload au lieu de spécifier explicitement toutes les options dans le composant. Lorsque cette case est cochée, l'option **Data file** ainsi que les autres options liées à l'utilitaire gpload ne s'appliquent plus. Veuillez vous référer au manuel du gpload de Greenplum pour plus d'informations concernant la création d'un fichier de contrôle.
	Control file	Saisissez le chemin d'accès au fichier de contrôle à utiliser, entre guillemets doubles, ou cliquez sur le bouton **[...]** afin de parcourir votre répertoire jusqu'au fichier de contrôle. Cette option est passée à l'utilitaire gpload via l'argument *-f*.
	CSV mode	Cochez cette case pour inclure des paramètres spécifiques au format CSV, tels que **Escape char** et **Text enclosure**.
	Field separator	Caractère, chaîne de caractères ou expression régulière utilisé pour séparer les champs. *Argument delim de l'utilitaire gpload. La valeur par défaut est |. Pour améliorer les performances, utilisez la valeur par défaut.*
	Escaped char	Caractère d'échappement de la ligne.
	Text enclosure	Caractères utilisés pour entourer le texte.
	Header (skips the first row of data file)	Cochez cette case pour ignorer la première ligne du fichier de données.
	Additional options	Définissez les arguments gpload dans le tableau correspondant. Cliquez sur le bouton **[+]** autant de fois que nécessaire afin d'ajouter les arguments au tableau. Cliquez sur le champ **Parameter** et choisissez l'argument souhaité dans la liste, puis cliquez sur le champ **Value** correspondant et saisissez une valeur entre guillemets.
		LOCAL_HOSTNAME : Nom de l'hôte ou adresse IP de la machine sur laquelle gpload fonctionne. Si cette machine est configurée avec plusieurs cartes réseau, vous pouvez spécifier le nom de l'hôte ou l'adresse IP de chaque carte réseau, afin de permettre au trafic d'utiliser simultanément toutes les cartes réseau. Par défaut, l'hôte local ou adresse IP principal(e) est utilisé(e).
		PORT (gpfdist port) : Numéro du port spécifique devant être utilisé par le programme de distribution du fichier gpfdist. Vous pouvez également renseigner **PORT_RANGE** afin de sélectionner un port disponible dans l'écart spécifié. Si **PORT** et **PORT_RANGE** sont spécifiés, **PORT** prévaut. Si aucun des deux n'est défini, un port disponible entre 8000 et 9000 est sélectionné par défaut. Si plusieurs noms d'hôtes sont déclarés dans **LOCAL_HOSTNAME**, le numéro du port est utilisé pour tous les hôtes. Cette configuration est souhaitée si vous voulez utiliser toutes les cartes réseau pour charger le même fichier ou le même ensemble de fichiers dans un répertoire donné.
		PORT_RANGE : Peut être utilisé à la place de **PORT (gpfdist port)** afin de spécifier une plage de numéros de ports parmi lesquels gpload peut choisir un port disponible pour cette instance du programme de distribution du fichier gpfdist.

		NULL_AS : Chaîne de caractères représentant une valeur null. La valeur par défaut est `\N` en mode TEXT et une valeur vide sans guillemet en mode CSV. Tout élément de données source correspondant à cette chaîne de caractères sera considéré comme valeur null.
		FORCE_NOT_NULL : En mode CSV, traite chaque colonne spécifiée comme si elle était entourée de guillemets et n'était donc pas une valeur NULL. Pour la chaîne de caractères null par défaut en mode CSV (aucun caractère entre deux délimiteurs), cela crée des valeurs manquantes à évaluer comme des chaînes de caractères de longueur zéro.
		ERROR_LIMIT (2 or higher) : Active le mode d'isolation d'une ligne pour cette opération de chargement. Lorsqu'il est activé et que le nombre maximal d'erreurs n'est pas atteint pour les instances de segments Greenplum durant le traitement de l'entrée, toutes les lignes correctes sont chargées et les lignes ayant des erreurs de format sont rejetées ou enregistrées dans la table **ERROR_TABLE**, si elle est disponible. Si le nombre limite d'erreurs est atteint, les lignes d'entrée ayant des erreurs de format arrêtent l'opération de chargement. Notez que l'isolation d'une ligne en erreur s'applique uniquement aux lignes de données ayant des erreurs de format, par exemple, des attributs manquants ou supplémentaires, des attributs d'un mauvais type de données ou des séquences d'encodage client invalides. Des erreurs de contraintes, telles que les violations de clé primaire, provoquent toujours l'arrêt du chargement. Lorsque cette option n'est pas activée, l'opération de chargement s'arrête à la première erreur rencontrée.
		ERROR_TABLE : Lorsque la limite **ERROR_LIMIT** est déclarée, spécifie une table d'erreur dans laquelle les lignes contenant des erreurs de format sont enregistrées lors de l'exécution en mode isolation d'une ligne. Vous pouvez observer cette table d'erreurs afin de voir les lignes en erreur n'ayant pas été chargées (s'il y en a).
	Log file	Parcourez votre système ou saisissez le chemin d'accès à votre fichier de log.
	Encoding	Saisissez le type d'encodage dans le champ.
	Specify gpload path	Cochez cette case afin de spécifier le chemin d'accès complet à l'exécutable gpload. Vous devez cocher cette case si le chemin d'accès n'est pas spécifié dans la variable d'environnement PATH.
	Full path to gpload executable	Chemin d'accès complet à l'exécutable gpload sur la machine en cours d'utilisation. Il est recommandé de spécifier le chemin d'accès dans la variable d'environnement PATH au lieu de sélectionner cette option.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **GPLOAD_OUTPUT** : les informations de sortie lors de l'exécution de l'utilitaire gpload. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est principalement utilisé lorsque des transformations particulières ne sont pas requises sur les données à charger dans la base de données. Ce composant peut être utilisé en standalone ou en tant que composant de sortie.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *Scénario : Insérer des données dans une base MySQL* du composant **tMysqlOutputBulkExec**.

tGreenplumInput

Propriétés du tGreenplumInput

Famille de composant	Databases/ Greenplum	
Fonction	Le composant **tGreenplumInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tGreenplumInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case lorsque vous utilisez le composant **tGreenplumConnection**.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Use cursor*	Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d'optimiser les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Greenplum.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé, consultez les scénarios des composants **tDBInput** dans :

- *Scénario: Mapper les données à l'aide d'une jointure implicite simple* .
- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tGreenplumOutput

Propriétés du tGreenplumOutput

Famille de composant	Databases/ Greenplum	
Fonction	Le composant **tGreenplumOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	**Le tGreenplumOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore.

		Create table if doesn't exist : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les

		colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tPostgresqlOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Greenplum. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé au composant **tGreenplumOutput** consultez :

- *Scénario: Mapper les données à l'aide d'une jointure implicite simple* .
- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du **tMysqlOutput**.

tGreenplumOutputBulk

Propriétés du tGreenplumOutputBulk

Les composants **tGreenplumOutputBulk** et **tGreenplumBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tGreenplumOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/Greenplum	
Fonction	Ecrit un fichier composé de colonnes et basé sur le délimiteur défini et sur les standards Greenplum.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Greenplum.	
Basic settings	*Property type*	Either **Built-in** or **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer. *Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.*
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Row separator*	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tGreenplumBulkExec**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données Greenplum.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé au **tGreenplumOutputBulk**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tGreenplumOutputBulkExec

Propriétés du tGreenplumOutputBulkExec

Les composants **tGreenplumOutputBulk** et **tGreenplumBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tGreenplumOutputBulkExec**.

Famille de composant	Databases/Greenplum	
Fonction	Le **tGreenplumOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tGreenplumOutputBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Greenplum.	
Basic settings	*Property Type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de la base de données. Seuls *localhost*, *127.0.0.1* ou l'adresse IP exacte de la machine locale permettent un fonctionnement optimal. Le serveur de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tGreenplumOutputBulkExec**.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer un nouvelle. **Clear a table** : supprimer le contenu de la table.

	File Name	Non du fichier à générer et à charger. *Ce fichier est généré sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Action on data*	Sélectionnez l'opération que vous voulez effectuer : **Bulk insert** ou **Bulk update**. Les informations demandées seront différentes en fonction de l'action choisie.
	Copy the OID for each row	Récupère les identifiants d'objet pour chaque ligne.
	Contains a header line with the names of each column in the file	Spécifiez que la table contient des lignes d'en-tête.
	File type	Sélectionnez le type de fichiers à traiter.
	Null string	Chaîne de caractères affichée pour signifier que la valeur est nulle.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs.
	Escaped char	Caractère d'échappement de la ligne.
	Text enclosure	Caractères utilisés pour entourer le texte.
	Force not null for columns	Définissez la nullabilité des colonnes **Force not null** : Cochez la case correspondant à la colonne (**Column**) que vous souhaitez définir comme n'étant pas nulle.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	

Limitation	Le serveur de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tGreenplumOutputBulkExec**, afin que le composant fonctionne correctement.

Scénarios associés

Pour un scénario associé au **tGreenplumOutputBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tGreenplumRollback

Propriétés du tGreenplumRollback

Famille de composant	Databases/Greenplum	
Fonction	Le **tGreenplumRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tGreenplumConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en association avec des composants Greenplum, notamment avec le **tGreenplumConnection** et le **tGreenplumCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tGreenplumRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMysqlRollback**.

tGreenplumRow

Propriétés du tGreenplumRow

Famille de composant	Databases/ Greenplum	
Fonction	Le **tGreenplumRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tGreenplumRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à lire.
	Query type	Peut être Built-in ou Repository.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tGreenplumSCD

Le composant **tGreenplumSCD** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tGreenplumSCD*.

tIngresBulkExec

Propriétés du tIngresBulkExec

Les composants **tIngresOutputBulk** et **tIngresBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tIngresOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Famille de composant	Databases/Ingres	
Fonction	Le composant **tIngresBulkExec** effectue une action d'insertion (INSERT) sur les données fournies.	
Objectif	Le **tIngresBulkExec** vous permet d'insérer des données en masse dans une table du SGBD Ingres, afin d'optimiser les performances.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Table	Nom de la table à remplir.
	VNode	Nom du nœud virtuel.
	Database	Nom de la base de données.
	Action on table	Actions pouvant être effectuées sur la table : **None** : Aucune opération. **Truncate** : Supprime toutes les lignes de la table et rend l'espace du fichier au système d'exploitation.
	File name	Nom du fichier à charger. *Ce fichier doit se situer sur la même machine que le serveur de la base de données.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets.

		Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Delete Working Files After Use	Cochez cette case pour supprimer les fichiers créés durant l'exécution.
Advanced settings	*Field Separator*	Saisissez le caractère, la chaîne de caractères ou l'expression régulière permettant de séparer les champs.
	Row Separator	Chaîne de caractères (ex : "\n" sous Unix) permettant de séparer les lignes.
	Null Indicator	Valeur de l'indicateur null.
	Session User	Utilisateur de la session définie (connexion à la base de données).
	Rollback	Activer ou désactiver le rollback.
	On Error	Politique de gestion d'erreur : **Continue** : Continuer l'exécution. **Terminate** : Terminer l'exécution.
	Reject Row File	Chemin d'accès et nom du fichier contenant les lignes rejetées. Disponible lorsque l'option **Continue** est sélectionnée dans la liste **On Error**.
	Error Count	Nombre d'erreurs avant arrêt de l'exécution. Disponible lorsque l'option **Terminate** est sélectionnée dans la liste **On Error**.
	Allocation	Nombre de pages initialement allouées à la table ou à l'index.
	Extend	Nombre de pages étendant la table ou l'index.
	Fill Factor	Spécifiez le pourcentage (de 1 à 100) de chaque page principale de données devant être remplie de lignes, dans des conditions idéales. Par exemple, si vous spécifiez un facteur de remplissage de 40, le serveur de SGBD remplit de lignes 40% de chaque page principale dans la table restructurée.
	Min Pages/Max Pages	Spécifiez le nombre minimum/maximum de pages principales qu'une table de hachage doit posséder. Ces nombres doivent être de 1 minimum.
	Leaf Fill	Une copie de masse peut spécifier une valeur leaffill. Cette clause spécifie le pourcentage (de 1 à 100) de chaque index de page de table B-tree devant être remplie par des lignes durant la copie. Cette clause peut être utilisée uniquement sur des tables ayant une structure de stockage B-tree.
	Non Leaf Fill	Une copie de masse peut spécifier une valeur nonleaffill. Cette clause spécifie le pourcentage (de 1 à 100) de chaque index de page de table B-tree ne devant pas être rempli par des lignes durant la copie. Cette clause peut être utilisée uniquement sur des tables ayant une structure de stockage B-tree.

	Row Estimate	Spécifiez le nombre estimé de lignes à copier d'un fichier à une table durant l'opération de copie de masse.
	Trailing WhiteSpace	Cochée par défaut, cette case est conçue pour supprimer les espaces en fin de champ et s'applique uniquement aux types de données VARCHAR, NVARCHAR et TEXT.
	Output	Emplacement où écrire le message d'erreur : **to console** : Ecrire le message dans la console. **to global variable** : Ecrire le message dans une variable globale.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE_DATA** : nombre de lignes lues. Cette variable est une variable **After** et retourne un entier. **NB_LINE_BAD** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le composant **tIngresBulkExec** est généralement utilisé avec le **tIngresConnection** et le **tIngresRow**. Il permet d'alimenter en masse une base de données Ingres, afin d'optimiser les performances.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Le serveur/client de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tIngresBulkExec**, afin que le composant fonctionne correctement. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez :

- *Scénario : Charger des données dans une table du SGBD Ingres*.

tIngresClose

Propriétés du tIngresClose

Famille de composant	Databases/Ingres	
Fonction	Le composant **tIngresClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Ingres.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tIngresConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tIngresConnection** et **tIngresRollback**. Ils sont généralement utilisés avec un composant **tIngresConnection** car il permet d'ouvrir une connexion pour la transaction en cours.

Pour un scénario associé au composant **tIngresClose**, consultez*tMysqlConnection*

tIngresCommit

Propriétés du tIngresCommit

Famille de composant	Databases/Ingres	
Fonction	Le composant **tIngresCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	Sélectionnez le composant **tIngresConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tIngresCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants JDBC et notamment avec les composants **tIngresConnection** et **tIngresRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tIngresCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tIngresConnection

Propriétés du tIngresConnection

Famille de composant	Databases/Ingres	
Fonction	Le composant **tIngresConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Server	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

		Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants JDBC et notamment avec les composants **tIngresCommit** et **tIngresRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au **tIngresConnection**, consultez *Scénario : Charger des données dans une table du SGBD Ingres*.

tIngresInput

Propriétés du tIngresInput

Famille de composant	Databases/ Ingres	
Fonction	Le composant **tIngresInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tIngresInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Server	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur de Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur de Studio Talend*.

		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Trim all the String/Char columns*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Ingres.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Consultez les scénarios associés au composant **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tIngresOutput

Propriétés du tIngresOutput

Famille de composant	Databases/Ingres	
Fonction	Le composant **tIngresOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tIngresOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire.

		Drop a table if exists and create : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions

		sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tIngresOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Ingres. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tIngresOutputBulk

Propriétés du tIngresOutputBulk

Les composants **tIngresOutputBulk** et **tIngresBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tIngresOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/Ingres	
Fonction	Le composant **tIngresOutputBulk** prépare un fichier avec le schéma défini et les données provenant du composant précédent.	
Objectif	Ce composant prépare le fichier dont les données sont insérées en masse dans le SGBD Ingres afin d'optimiser les performances.	
Basic settings	*Property Type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer. *Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.*
	Append the File	Cochez cette case pour ajouter de nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Field Separator*	Saisissez le caractère, la chaîne de caractères ou l'expression régulière permettant de séparer les champs.
	Row Separator	Chaîne de caractères (ex : "\n" sous Unix) permettant de séparer les lignes.
	Include Header	Cochez cette case pour inclure l'en-tête de colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le composant **tIngresOutputBulk** est généralement utilisé avec le **tIngresBulkExec**, afin de sauvegarder des données entrantes dans un fichier. Ces données sont ensuite insérées en masse dans une base de données Ingres **tIngresBulkExec** afin d'optimiser les performances.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez :

- *Scénario : Charger des données dans une table du SGBD Ingres*,

tIngresOutputBulkExec

tIngresOutputBulkExec

Les composants **tIngresOutputBulk** et **tIngresBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tIngresOutputBulkExec**.

Famille de composant	Databases/Ingres	
Fonction	Le composant **tIngresOutputBulkExec** prépare un fichier de sortie et l'utilise pour alimenter une table dans le SGBD Ingres.	
Objectif	Le **tIngresOutputBulkExec** insère des données en masse dans le SGBD Ingres afin d'optimiser les performances.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Table	Nom de la table à remplir.
	VNode	Nom du nœud virtuel. Le serveur de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tIngresOutputBulkExec**.
	Database	Nom de la base de données.
	Action on table	Actions pouvant être effectuées sur la table : **None** : Aucune opération. **Truncate** : Supprime toutes les lignes de la table et rend l'espace du fichier au système d'exploitation.
	File name	Nom du fichier à générer et à charger. *Ce fichier est généré sur la machine spécifiée dans le champ **VNode** et doit être sur la même machine que le serveur de la base de données.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous

		que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Delete Working Files After Use	Cochez cette case pour supprimer les fichiers créés durant l'exécution.
Advanced settings	*Field Separator*	Saisissez le caractère, la chaîne de caractères ou l'expression régulière permettant de séparer les champs.
	Row Separator	Chaîne de caractères (ex : "\n" sous Unix) permettant de séparer les lignes.
	On Error	Politique de gestion d'erreur : **Continue** : Continuer l'exécution. **Terminate** : Terminer l'exécution.
	Reject Row File	Chemin d'accès et nom du fichier contenant les lignes rejetées. Disponible lorsque l'option **Continue** est sélectionnée dans la liste **On Error**.
	Error Count	Nombre d'erreurs avant arrêt de l'exécution. Disponible lorsque l'option **Terminate** est sélectionnée dans la liste **On Error**.
	Rollback	Activer ou désactiver le rollback.
	Null Indicator	Valeur de l'indicateur null.
	Session User	Utilisateur de la session définie (connexion à la base de données).
	Allocation	Nombre de pages initialement allouées à la table ou à l'index.
	Extend	Nombre de pages étendant la table ou l'index.
	Fill Factor	Spécifiez le pourcentage (de 1 à 100) de chaque page principale de données devant être remplie de lignes, dans des conditions idéales. Par exemple, si vous spécifiez un facteur de remplissage de 40, le serveur de SGBD remplit de lignes 40% de chaque page principale dans la table restructurée.
	Min Pages/Max Pages	Spécifiez le nombre minimum/maximum de pages principales qu'une table de hachage doit posséder. Ces nombres doivent être de 1 minimum.
	Leaf Fill	Une copie de masse peut spécifier une valeur leaffill. Cette clause spécifie le pourcentage (de 1 à 100) de chaque index de page de table B-tree devant être remplie par des lignes durant la copie. Cette clause peut être utilisée uniquement sur des tables ayant une structure de stockage B-tree.
	Non Leaf Fill	Une copie de masse peut spécifier une valeur nonleaffill. Cette clause spécifie le pourcentage (de 1 à 100) de chaque index de page de table B-tree ne devant pas être rempli par des lignes durant la

		copie. Cette clause peut être utilisée uniquement sur des tables ayant une structure de stockage B-tree.
	Row Estimate	Spécifiez le nombre estimé de lignes à copier d'un fichier à une table durant l'opération de copie de masse.
	Trailing WhiteSpace	Cochée par défaut, cette case est conçue pour supprimer les espaces en fin de champ et s'applique uniquement aux types de données VARCHAR, NVARCHAR et TEXT.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	Output	Emplacement où écrire le message d'erreur : **to console** : Ecrire le message dans la console. **to global variable** : Ecrire le message dans une variable globale.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Le composant **tIngresOutputBulkExec** est généralement utilisé avec le **tIngresConnection** et le **tIngresRow**. Il prépare un fichier de sortis et l'alimente en masse avec des données dans le SGBD Ingres, afin d'optimiser les performances.	
Limitation	Le serveur/client de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tIngresOutputBulkExec**, afin que le composant fonctionne correctement. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Charger des données dans une table du SGBD Ingres

Dans ce scénario, un composant **tIngresOutputBulkExec** est utilisé afin de préparer un fichier de sortie contenant des données provenant d'un fichier .csv concernant des employés et utilise ensuite ce fichier de sortie afin d'alimenter une table dans une base de données Ingres.

Déposer et relier les composants

1. Déposez un **tIngresConnection**, un **tFileInputDelimited** et un **tIngresOutputBulkExec** de la **Palette** dans l'espace de modélisation graphique.
2. Renommez le **tIngresOutputBulkExec** en **save_a_copy_and_load_to_DB**.
3. Reliez le **tIngresConnection** au **tFileInputDelimited** à l'aide d'un lien **OnSubjobOk**.
4. Reliez le **tFileInputDelimited** au **tIngresOutputBulkExec** à l'aide d'un lien **Row > Main**.

Configurer les composants

1. Double-cliquez sur le **tIngresConnection** pour ouvrir sa vue **Component**.

2. Dans le champ **Server**, saisissez l'adresse du serveur où se trouve le SGBD Ingres, par exemple *"localhost"*.

   Laissez la valeur du champ **Port** par défaut.

3. Dans le champ **Database**, saisissez le nom de la base de données Ingres, par exemple *"research"*.

4. Dans les champs **Username** et **Password**, saisissez les informations d'authentification.

   Une variable de contexte est utilisée pour le mot de passe. Pour plus d'informations concernant les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

5. Double-cliquez sur le composant **tFileInputDelimited** pour ouvrir sa vue **Component**.

6. Sélectionnez le fichier source en cliquant sur le bouton **[...]** à côté du champ **File name/Stream** et en parcourant votre système.

7. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir l'éditeur de schéma.

8. Cliquez quatre fois sur le bouton **[+]** pour ajouter quatre colonnes, par exemple *name*, *age*, *job* et *dept*, dont le type de données est respectivement *string*, *Integer*, *string* et *string*.

   Cliquez sur **OK** pour fermer l'éditeur de schéma.

   Cliquez sur **Yes** dans la fenêtre qui s'ouvre afin d'accepter la propagation des modifications au composant suivant.

   Laissez les autres composants tels qu'ils sont.

9. Double-cliquez sur le **tIngresOutputBulkExec** pour ouvrir sa vue **Component**.

10. Dans le champ **Table**, saisissez le nom de la table pour l'insertion de données.

11. Dans les champs **VNode** et **Database**, saisissez le nom du nœud virtuel et de la base de données.

12. Dans le champ **File Name**, saisissez le chemin d'accès complet au fichier qui contiendra les données du fichier source.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** pour exécuter le Job.

Comme affiché ci-dessus, les données concernant les employés sont écrites dans la table *employee* dans la base de données *research*, sur le nœud *talendbj*. Le fichier de sortie *employee_research.csv* a été généré à l'emplacement spécifié, *C:/Users/talend/Desktop*.

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC*,
- *Scénario 1 : Ajouter une colonne et modifier les données*.

tIngresRollback

Propriétés du tIngresRollback

Famille de composant	Databases/Ingres	
Fonction	Le **tIngresRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tIngresConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en association avec des composants Ingres, notamment avec le **tIngresConnection** et le **tIngresCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tIngresRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMysqlRollback**.

tIngresRow

Propriétés du tIngresRow

Famille de composant	Databases/ Ingres	
Fonction	**tIngresRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tIngresRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à traiter.
	Query type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans	

	l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tIngresSCD

Le composant **tIngresSCD** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tIngresSCD*.

tNetezzaBulkExec

Propriétés du tNetezzaBulkExec

Famille de composant	Databases/Netezza	
Fonction	Le composant **tNetezzaBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tNetezzaBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Netezza.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit exister pour que l'opération d'Insert soit autorisée.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	File Name	Nom du fichier à charger. Ce fichier doit se situer sur la même machine que le *Studio Talend* ou le Job contenant un **tNetezzaBulkExec**, afin que le composant fonctionne correctement.
Advanced settings	*Field Separator*	Caractère, chaîne ou expression régulière séparant les champs.
	Escape character	Caractère d'échappement de la ligne.
	Date format / Date delimiter	**Date format**: Utilisez ce champ pour représenter l'ordre des mois, des jours et des années sous forme de chaîne de caractères. **Date delimiter** : Utilisez ce champ pour définir l'élément séparateur entre les valeurs.
	Time format/ Time delimiter	**Time format** : Utilisez ce champ pour représenter le format de l'heure sous forme de chaîne de caractères. **Time delimiter** : Utilisez ce champ pour définir l'élément séparateur entre les valeurs.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Max Errors	Saisissez le nombre maximal d'erreurs possibles avant que le processus ne s'arrête.
	Skip Rows	Renseignez le nombre de lignes à ignorer.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est principalement utilisé lorsque des transformations non particulières sont requises sur les données à charger dans la base de données Netezza.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour des scénarios associés au composant **tNetezzaBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tNetezzaClose

Propriétés du tNetezzaClose

Famille de composant	Databases/Netezza	
Fonction	Le composant **tNetezzaClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Netezza.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tNetezzaConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tNetezzaConnection** et **tNetezzaRollback**. Il est généralement utilisé avec un composant **tNetezzaConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tNetezzaClose**, consultez *tMysqlConnection*.

tNetezzaCommit

Propriétés du tNetezzaCommit

Famille de composant	Databases/Netezza	
Fonction	Le composant **tNetezzaCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tNetezzaConnection** dans la liste.
	Close Connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tNetezzaCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Netezza et notamment avec les composants **tNetezzaConnection** et **tNetezzaRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tNetezzaCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tNetezzaConnection

Propriétés du tNetezzaConnection

Famille de composant	Databases/Netezza	
Fonction	Le composant **tNetezzaConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Additional JDBC Parameters	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

		Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants Netezza, notamment les composants **tNetezzaCommit** et **tNetezzaRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au composant **tNetezzaConnection**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tNetezzaInput

Propriétés du tNetezzaInput

Famille de composant	Databases/ Netezza	
Fonction	Le composant **tNetezzaInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tNetezzaInput** exécute une requête de base de données dans laquelle l'ordre doit strictement correspondre à l'ordre défini dans le schéma. Puis la liste des champs est transmise au composant via une connexion Main row.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionne le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table name	Nom de la table de base de données.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Use cursor*	Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d'optimiser les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Netezza.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Consultez les scénarios du composant **tNetezzaInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Pour un scénario associé, consultez *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tNetezzaNzLoad

Ce composant invoque l'utilitaire nzload de Netezza afin d'insérer des enregistrements dans une base de données Netezza. Ce composant peut être utilisé soit en mode standalone, il chargera alors les données à partir d'un fichier de données existant, soit connecté à un flux d'entrée, afin de charger des données à partir du composant connecté.

Propriétés du tNetezzaNzLoad

Famille de composant	Databases/Netezza	
Fonction	Le composant **tNetezzaNzLoad** insère des données dans la table d'une base de données Netezza via l'utilitaire nzload de Netezza.	
Objectif	Ce composant permet de charger en masse des données dans une table Netezza soit à partir d'un fichier, soit à partir d'un flux d'entrée, soit à partir d'un tube nommé.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table dans laquelle les données doivent être insérées.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Data file	Chemin d'accès complet au fichier de données à utiliser. Si ce composant est utilisé seul (non connecté à un autre composant par un flux d'entrée), alors saisissez le nom d'un fichier existant à charger dans la base de données. S'il est connecté à un autre composant par un flux d'entrée, saisissez le nom du fichier à générer et à écrire avec les données d'entrée, afin de l'utiliser plus tard avec nzload pour charger dans la base de données.
	Use named-pipe	Cochez cette case afin d'utiliser un tube nommé à la place d'un fichier de données. Cette option ne peut être utilisée que lorsque le composant est connecté à un autre composant par un flux d'entrée. Quand la case est cochée, aucun fichier de données n'est généré, et les données sont transférées à l'utilitaire nzload via un tube nommé. Cette option améliore grandement les performances sous Linux et Windows. Ce composant, en mode tube nommé, utilise une interface JNI pour créer un tube nommé et écrire dedans, dans toute plateforme Windows. Le chemin d'accès associé au fichier DLL JNI doit être configuré dans le chemin d'accès à la bibliothèque Java. Le composant est automatiquement fourni dans le *Studio Talend* avec deux DLL pour les systèmes d'exploitation de 32 et 64 bits.
	Named-pipe name	Spécifiez un nom pour le tube nommé à utiliser. Vérifiez que le nom saisi est valide.
Advanced settings	*Use existing control file*	Cochez cette case pour fournir un fichier de contrôle à utiliser avec l'utilitaire nzload au lieu de spécifier explicitement toutes les options dans le composant. Lorsque cette case est cochée, l'option **Data file** ainsi que les autres options liées au nzload ne s'appliquent plus. Veuillez vous référer au manuel du nzload de Netezza pour plus d'informations concernant la création d'un fichier de contrôle.
	Control file	Saisissez le chemin d'accès au fichier de contrôle à utiliser, entre guillemets doubles, ou cliquez sur le bouton **[...]** afin de parcourir votre répertoire jusqu'au fichier de contrôle. Cette option est passée à l'utilitaire nzload via l'argument *-cf*.
	Field separator	Caractère, chaîne de caractères ou expression régulière utilisé pour séparer les champs. *Argument delim de l'utilitaire nzload. Si vous n'utilisez pas l'option **Wrap quotes around fields**, vous devez être sûr que le délimiteur n'est pas inclus dans les données insérées dans la base de données. La valeur par défaut est \t ou TAB. Pour améliorer les performances, utilisez la valeur par défaut.*
	Wrap quotes around fields	Cette option ne s'applique qu'aux colonnes de type *String*, *Byte*, *Byte[]*, *Char*, et *Object*. Sélectionnez soit : **None** : n'entourez pas les valeurs des colonnes de guillemets.

		Single quote : entourez les valeurs des colonnes de guillemets simples. **Double quote** : entourez les valeurs des colonnes de guillemets doubles. *Si vous utilisez l'option **Single quote** ou **Double quote**, il est nécessaire d'utiliser \ en tant que **Escape char**.*
	Advanced options	Définissez les arguments nzload dans le tableau correspondant. Cliquez sur le bouton **[+]** autant de fois que nécessaire afin d'ajouter les arguments au tableau. Cliquez sur le champ **Parameter** et choisissez l'argument souhaité dans la liste, puis cliquez sur le champ **Value** correspondant et saisissez une valeur entre guillemets.
Parameter	*-If*	Nom du fichier de log à générer. Les logs seront ajoutés à la suite du fichier de log, s'il existe déjà. Si le paramètre n'est pas spécifié, le nom par défaut du fichier de log est '*<table_name>.<db_name>.nzlog*'. Il est généré dans le répertoire de travail courant dans lequel le Job est exécuté.
	-bf	Nom du fichier de mauvais enregistrements à générer. Le fichier de mauvais enregistrements contient tous les enregistrements ne pouvant être chargés, à cause d'une erreur interne Netezza. Les enregistrements seront ajoutés à la suite si le fichier de mauvais enregistrements existe déjà. Si le paramètre n'est pas spécifié, le nom par défaut du fichier de mauvais enregistrements est '*<table_name>.<db_name>.nzbad*'. Il est généré dans le répertoire de travail courant, dans lequel le Job est exécuté.
	-ouputDir	Chemin d'accès au répertoire dans lequel les fichiers de log et de mauvais enregistrements sont générés. Si le paramètre n'est pas spécifié, les fichiers seront générés sous le répertoire courant dans lequel le Job est exécuté.
	-logFileSize	Taille maximale du fichier de log. La valeur est en MB. La valeur par défaut est de 2000 ou 2GB. Pour économiser de l'espace disque, spécifiez un petite valeur si votre Job est souvent exécuté.
	-compress	Sélectionnez cette option si le fichier de données est compressée. Les valeurs valides sont "**TRUE**" et "**FALSE**". La valeur par défaut est "**FALSE**". Cette option n'est valide que si ce composant est utilisé par lui-même et non connecté à un autre composant via un flux d'entrée.
	-skipRows <n>	Nombre de lignes à ignorer au début du fichier de données. Paramétrez la valeur à "1" si vous souhaitez ignorer la ligne d'en-tête du fichier d'entrée. La valeur par défaut est "0". Cette option ne doit être sélectionnée que si le composant est utilisé par lui-même et non connecté à un autre composant via un flux d'entrée.
	-maxRows <n>	Nombre de lignes à charger à partir du fichier d'entrée. Cette option ne doit être sélectionnée que si le composant est utilisé par lui-même et non connecté à un autre composant via un flux d'entrée.
	-maxErrors	Nombre maximal d'enregistrements d'erreurs à autoriser avant de terminer le processus de chargement. La valeur par défaut est "1".
	-ignoreZero	Les zéros contenus dans une source binaire généreront des erreurs. Paramétrez cette option à "NO" pour générer des erreurs ou à "YES" pour ignorer les bits ayant la valeur zéro. La valeur par défaut est "NO".
	-requireQuotes	Cette option nécessite que toutes les valeurs soient entourées de guillemets. La valeur par défaut est "FALSE".

		Cette option ne fonctionne pas avec un flux d'entrée pour le moment. Utilisez cette option uniquement en mode standalone avec un fichier existant.
	-nullValue <token>	Spécifiez le jeton pour indiquer une valeur null dans le fichier de données. La valeur par défaut est "NULL". Afin d'améliorer légèrement les performances, vous pouvez configurer cette valeur comme un champ vide en spécifiant la valeur : "\\"".
	-fillRecord	Traite les derniers champs d'entrée manquants comme des null. Vous n'avez pas besoin de spécifier de valeur pour cette option dans le champ du tableau. Cette option n'est pas activée par défaut, les champs d'entrée devront donc correspondre exactement à toutes les colonnes de la table par défaut. Les derniers champs d'entrée doivent être nullables dans la base de données.
	-ctrlChar	Accepte les caractères de contrôle dans les champs char/varchar (vous devrez échapper les caractères NUL, CR et LF). Vous n'avez pas besoin de spécifier de valeur pour cette option dans le champ de valeur du tableau. Cette option est désactivée par défaut.
	-ctInString	Accepte le caractère CR non échappé dans des champs char/varchar (LF devient une fin de ligne). Vous n'avez pas besoin de spécifier de valeur pour cette option dans le champ de valeur du tableau. Cette option est désactivée par défaut.
	-truncString	Tronque toute valeur de chaîne de caractères qui dépasse sa valeur déclarée de stockage char/varchar. Vous n'avez pas besoin de spécifier de valeur pour cette option dans le champ de valeur du tableau. Cette option est désactivée par défaut.
	-dateStyle	Spécifiez le format de la date des données d'entrée. Les valeurs valides sont : "YMD", "Y2MD", "DMY", "DMY2", "MDY", "MDY2", "MONDY", "MONDY2". La valeur par défaut est "YMD". Le format de date dans la colonne du schéma du composant doit correspondre à la valeur spécifiée ici. Par exemple, si vous souhaitez charger une colonne DATE, spécifiez ainsi le format de date dans le schéma du composant : "yyyy-MM-dd" et paramétrez l'option -dateStyle à "YMD". Pour une description plus précise du chargement de champs de dates et de temps, consultez *Charger des colonnes de type DATE, TIME et TIMESTAMP*.
	-dateDelim	Caractère délimiteur entre les parties d'une date. La valeur par défaut est "-" pour tous les styles de date, sauf pour "MONDY[2]" pour lequel elle est " " (espace vide). Le format de date dans la colonne du schéma du composant doit correspondre à la valeur spécifiée ici.
	-y2Base	Première année exprimable via l'utilisation du style de date à deux chiffres (Y2).
	-timeStyle	Spécifiez le format du temps dans les données d'entrée. Les valeurs valides sont : "24HOUR" et "12HOUR". La valeur par défaut est "24HOUR". Pour améliorer légèrement les performances, gardez la valeur par défaut. Le format de temps dans la colonne du schéma du composant doit correspondre à la valeur spécifiée ici. Par exemple, si vous souhaitez charger une colonne de type TIME, configurez le format de date dans le schéma du composant à "HH:mm:ss" et l'option -timeStyle à "24HOUR". Pour une description plus précise du chargement de champs de dates et de temps, consultez *Charger des colonnes de type DATE, TIME et TIMESTAMP*.

	-timeDelim	Caractère délimiteur entre les parties d'une donnée de temps. La valeur par défaut est ":". Le format de temps dans la colonne du schéma du composant doit correspondre à la valeur spécifiée ici.
	-timeRoundNanos	Autorise mais arrondit les nombres différents de zéro plus petits qu'une microseconde.
	-boolStyle	Spécifiez le format dans lequel les données de type Boolean sont écrites dans les données. Les valeurs valides sont : "1_0", "T_F", "Y_N", "TRUE_FALSE", "YES". La valeur par défaut est "1_0". Pour améliorer légèrement les performances, gardez la valeur par défaut.
	-allowRelay	Permet de continuer le chargement malgré une ou plusieurs réinitialisation(s) ou échec(s) des SPU (Snippet Processing Unit). Par défaut, ce comportement n'est pas autorisé.
	-allowRelay <n>	Spécifiez le nombre de continuations du chargement autorisé. La valeur par défaut est "1".
	Encoding	Sélectionnez l'encodage à partir de la liste.
	Specify nzload path	Cochez cette case afin de spécifier le chemin d'accès complet à l'exécutable nzload. Vous devez cocher cette case si le chemin d'accès n'est pas spécifié dans la variable d'environnement PATH.
	Full path to nzload executable	Chemin d'accès complet à l'exécutable nzload sur la machine en cours d'utilisation. Il est recommandé de spécifier le chemin d'accès dans la variable d'environnement PATH au lieu de sélectionner cette option.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est principalement utilisé lorsque des transformations particulières ne sont pas requises sur les données à charger dans la base de données. Ce composant peut être utilisé en standalone ou en tant que composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Charger des colonnes de type DATE, TIME et TIMESTAMP

Lorsque ce composant est utilisé avec un flux d'entrée, le format de date spécifié dans le schéma du composant doit correspondre à la valeur spécifiée pour les options -dateStyle, -dateDelim, -timeStyle, et -timeDelim. Veuillez vous référer aux exemples suivants :

Type de données	Format de date dans le schéma	-dateStyle	-dateDelim	-timeStyle	-timeDelim
DATE	"yyyy-MM-dd"	"YMD"	"-"	n/a	n/a
TIME	"HH:mm:ss"	n/a	n/a	"24HOUR"	":"
TIMESTAMP	"yyyy-MM-dd HH:mm:ss"	"YMD"	"-"	"24HOUR"	":"

Scénario associé

Pour un scénario associé, consultez *Scénario : Insérer des données dans une base MySQL* du composant **tMysqlOutputBulkExec**.

tNetezzaOutput

Propriétés du tNetezzaOutput

Famille de composant	Databases/Netezza	
Fonction	Le composant **tNetezzaOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tNetezzaOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **Default** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore.

		Create table if doesn't exist : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondant au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.

Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Use Batch size	Cochez cette case afin d'activer le mode batch pour le traitement des données. Dans le champ **Batch Size** qui apparaît, vous pouvez saisir le nombre de lignes par lot à charger. Cette case est disponible uniquement si vous avez sélectionné l'option **Insert**, **Update** ou **Delete** dans la liste **Action on data**.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tNetezzaOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Netezza. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé au composant **tNetezzaOutput**, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput.**
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tNetezzaRollback

Propriétés du tNetezzaRollback

Famille de composant	Databases/Netezza	
Fonction	Le **tNetezzaRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant permet d'éviter le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tNetezzaConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Netezza, notamment les composants **tNetezzaConnection** et **tNetezzaCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tNetezzaRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille*.

tNetezzaRow

Propriétés du tNetezzaRow

Famille de composant	Databases/ Netezza	
Fonction	**tNetezzaRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tNetezzaRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à traiter.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	NB_EFFECTED : nombre de lignes établies. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé au composant **tNetezzaRow**, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL*.

tNetezzaSCD

Le composant **tNetezzaSCD** appartient à deux familles de composants : Business Intelligence et Databases. Pour plus d'informations sur ce composant, consultez *tNetezzaSCD*.

tParAccelBulkExec

Propriétés du tParAccelBulkExec

Les composants **tParAccelOutputBulk** et **tParAccelBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tParAccelOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Famille de composant	Databases/ParAccel	
Fonction	Le composant **tParAccelBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tParAccelBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données ParAccel.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table.

		Drop and create table : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Copy mode*	Sélectionnez le mode de Copy que vous souhaitez utiliser. **Basic** : Mode classique, sans optimisation. **Parallel** : Mode permettant d'utiliser plusieurs API internes à ParAccel, afin d'améliorer les performances de chargement.
	Filename	Nom du fichier à charger. *Ce fichier est situé sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
	File Type	Sélectionnez dans la liste le type de fichier.
	Field Layout	Sélectionnez dans la liste la disposition du fichier.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs (si vous chargez un fichier délimité).
	Explicit IDs	L'ID sera assigné directement par la base de données ou est déjà présent dans le fichier à charger.
	Remove Quotes	Cochez cette case afin de retirer les guillemets de votre fichier à charger.
	Max. Errors	Saisissez le nombre maximal d'erreurs avant que votre Job ne s'arrête.
	Date Format	Saisissez le format de date qui sera utilisé.
	Time/Timestamp Format	Saisissez le format de date et heure qui sera utilisé.

	Additional COPY Options	Saisissez l'option spécifique à ParAccel et personnalisée que vous souhaitez utiliser.
	Log file	Saisissez le chemin d'accès à votre fichier de log, ou parcourez votre répertoire.
	Logging level	Sélectionnez le niveau de verbose à retourner.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec un composant **tParAccelOutputBulk**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données ParAccel.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour plus d'informations relatives au fonctionnement de **tParAccelBulkExec**, consultez les scénarios suivants dans :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du composant **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du composant **tOracleBulkExec**.

tParAccelClose

Propriétés du tParAccelClose

Famille de composant	Databases/ParAccel	
Fonction	Le composant **tParAccelClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données ParAccel.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tParAccelConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tParAccelConnection** et **tParAccelRollback**. Il est généralement utilisé avec un composant **tParAccelConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tParAccelClose**, consultez *tMysqlConnection*.

tParAccelCommit

Propriétés du tParAccelCommit

Famille de composant	Databases/ParAccel	
Fonction	Le composant **tParAccelCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tParAccelConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** >**Main** pour relier le **tParAccelCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants ParAccel et notamment **tParAccelConnection** et **tParAccelRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tParAccelCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tParAccelConnection

Propriétés du tParAccelConnection

Famille de composant	Databases/ParAccel	
Fonction	Le composant **tParAccelConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

		Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants ParAccel, notamment les composants **tParAccelCommit** et **tParAccelRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au composant **tParAccelConnection**, consultez *tMysqlConnection*.

tParAccelInput

Propriétés du tParAccelInput

Famille de composant	Databases/ ParAccel	
Fonction	Le composant **tParAccelInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tParAccelInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table name	Nom de la table de base de données.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Use cursor*	Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d'optimiser les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données ParAccel.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez les scénarios du composant **tParAccelInput** :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

tParAccelOutput

Propriétés du tParAccelOutput

Famille de composant	Databases/ ParAccel	
Fonction	Le composant **tParAccelOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	**Le tParAccelOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore.

		Create table if doesn't exist : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les

		colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tParAccelOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données ParAccel. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé au composant **tParAccelOutput** consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du **tMysqlOutput**.

tParAccelOutputBulk

Propriétés du tParAccelOutputBulk

Les composants **tParAccelOutputBulk** et **tParAccelBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tParAccelOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/ParAccel	
Fonction	Ecrit un fichier composé de colonnes et basé sur le délimiteur défini et sur les standards ParAccel.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données ParAccel.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer. *Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.*
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Row separator*	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tParAccelBulkExec**. Ensemble ils offrent un gain de performance important pour l'alimentation d'une base de données ParAccel.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé au composant **tParAccelOutputBulk**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tParAccelOutputBulkExec

Propriétés du tParAccelOutputBulkExec

Les composants **tParAccelOutputBulk** et **tParAccelBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tParAccelOutputBulkExec**.

Famille de composant	Databases/ParAccel	
Fonction	Le **tParAccelOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tParAccelOutputBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données ParAccel.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de la base de données. Seuls *localhost*, *127.0.0.1* ou l'adresse IP exacte de la machine locale permettent un fonctionnement optimal. Le serveur de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tParAccelOutputBulkExec**.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None**: n'effectuer aucune opération de table. **Drop and create table**: supprimer la table puis en créer une nouvelle. **Create table**: créer une table qui n'existe pas encore. **Create table if not exists**: créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table**: supprimer le contenu de la table.

	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Filename	Nom du fichier à générer et à charger. *Ce fichier est généré sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
Advanced settings	*Copy mode*	Sélectionnez le mode de Copy que vous souhaitez utiliser. **Basic** : Mode classique, sans optimisation. **Parallel** : Mode permettant d'utiliser plusieurs API internes à ParAccel, afin d'améliorer les performances de chargement.
	Filename	Nom du fichier à traiter et chemin d'accès.
	File Type	Sélectionnez dans la liste le type de fichier.
	Field Layout	Sélectionnez dans la liste la disposition du fichier.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs (si vous chargez un fichier délimité).
	Explicit IDs	L'ID sera assigné directement par la base de données ou est déjà présent dans le fichier à charger.
	Remove Quotes	Cochez cette case afin de retirer les guillemets de votre fichier à charger.
	Max. Errors	Saisissez le nombre maximal d'erreurs avant que votre Job ne s'arrête.
	Date Format	Saisissez le format de date qui sera utilisé.
	Time/Timestamp Format	Saisissez le format de date et heure qui sera utilisé.
	Additional COPY Options	Saisissez l'option spécifique à ParAccel et personnalisée que vous souhaitez utiliser.

	Log file	Saisissez le chemin d'accès à votre fichier de log, ou parcourez votre répertoire.
	Logging level	Sélectionnez le niveau de verbose à retourner.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	
Limitation	Le serveur de la base de données doit être installé sur la même machine que le *Studio Talend* ou que le Job contenant un **tParAccelOutputBulkExec**, afin que le composant fonctionne correctement.	

Scénarios associés

Pour un scénario associé au composant **tParAccelOutputBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tParAccelRollback

Propriétés du tParAccelRollback

Famille de composant	Databases/ParAccel	
Fonction	Le **tParAccelRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant permet d'éviter le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tParAccelConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants ParAccel, notamment les composants **tParAccelConnection** et **tParAccelCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tParAccelRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille*.

tParAccelRow

Propriétés du tParAccelRow

Famille de composant	Databases/ ParAccel	
Fonction	Le **tParAccelRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tParAccelRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table name	Nom de la table de base à lire.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tParAccelSCD

Le composant **tParAccelSCD** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tParAccelSCD*.

tRedshiftBulkExec

Propriétés du tRedshiftBulkExec

Les composants **tRedshiftOutputBulk** et **tRedshiftBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus pour charger des données dans Amazon Redshift à partir d'un fichier CSV/délimité dans Amazon S3. Au cours de la première étape, un fichier délimlité/CSV est généré. Au cours de la seconde étape, le fichier est utilisé dans une instruction INSERT utilisée pour alimenter. Cette exécution en deux étapes est unifiée dans le composant **tRedshiftOutputBulkExec**. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans Amazon Redshift.

Famille de composant	Databases/Amazon Redshift	
Fonction	Le **tRedshiftBulkExec** charge des données dans une table Amazon Redshift à partir d'un fichier plat situé dans Amazon S3.	
Objectif	Ce composant vous permet de charger des données dans à partir d'un fichier Amazon S3.	
Basic settings	*Property Type*	Peut être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Propriété utilisée ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
Database settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Saisissez l'adresse IP ou le nom de l'hôte du serveur de la base de données.
	Port	Saisissez le numéro du port d'écoute du serveur de la base de données.
	Database	Saisissez le nom de la base de données.
	Schema	Saisissez le nom du schéma.
	Username et *Password*	Saisissez les données d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table Name	Spécifiez le nom de la table à écrire. Notez que seule une table peut être écrite à la fois.
	Action on table	Sur la table définie, vous pouvez effectuer l'une des opérations suivantes : • **None** : n'effectuer aucune opération sur la table. • **Drop and create table** : supprimer la table puis en créer une nouvelle. • **Create table** : créer une table qui n'existe pas encore. • **Create table if not exists** : créer la table si nécessaire. • **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. • **Clear table** : supprimer le contenu de la table.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.

		Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
S3 Setting	*Access Key*	Spécifiez l'ID de la clé d'accès identifiant de manière unique un compte AWS. Pour plus d'informations concernant l'obtention de votre clé d'accès et clé secrète, consultez Getting Your AWS Access Keys (en anglais).
	Secret Key	Spécifiez la clé secrète d'accès, constituant les informations de sécurité, ainsi que la clé d'accès. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Bucket	Saisissez le nom du bucket Amazon S3 dans lequel le fichier est situé.
	Key	Saisissez la clé d'objet assignée au fichier à charger dans Amazon S3.
Advanced settings	*File type*	Sélectionnez dans la liste le type de fichier dans Amazon S3 : • **Delimited file or CSV** : fichier délimité/CSV. • **JSON** : fichier JSON. • **Fixed width** : fichier à taille fixe.
	Fields terminated by	Saisissez le caractère utilisé pour séparer les champs. Ce champ est disponible uniquement lorsque l'option **Delimited file or CSV** est sélectionnée dans la liste **File type**.
	Enclosed by	Sélectionnez le caractère par lequel entourer les champs. Cette liste est disponible uniquement lorsque l'option **Delimited file or CSV** est sélectionnée dans la liste **File type**.
	JSON mapping	Spécifiez comment mapper les éléments de données dans le fichier JSON source dans Amazon S3 vers les colonnes dans la table cible d'Amazon Redshift. Les valeurs valides sont : • **auto** : mappe les données en faisant correspondre les clés ou noms des objets dans les paires source nom/valeur vers les noms de colonnes dans la table cible. L'argument est sensible à la casse et doit être entouré de guillemets doubles. • *s3://jsonpaths_file* : mappe les données à l'aide du fichier nommé JSONPaths. Le paramètre doit être une clé d'objet Amazon S3 entouré de guillemets doubles et faisant référence à un fichier, par exemple, *s3://mybucket/jsonpaths.txt*. Pour plus d'informations, consultez http://docs.aws.amazon.com/redshift/latest/dg/r_COPY.html (en anglais). Ce champ est disponible uniquement lorsque l'option **JSON** est sélectionnée dans la liste **File type**.
	Fixed width mapping	Saisissez une chaîne de caractères spécifiant un libellé et une largeur de colonne personnalisés entre guillemets doubles. Le format de la chaîne de caractères est le suivant :

		`ColumnLabel1:ColumnWidth1,ColumnLabel2:ColumnWidth2,....` Notez que le libellé de la colonne dans la chaîne de caractères n'a aucun lien avec le nom de la colonne de la table et peut être une chaîne de caractères ou un entier. L'ordre des paires libellé/largeur doit correspondre à l'ordre exact des colonnes de la table. Ce champ est disponible uniquement lorsque l'option **Fixed width** est sélectionnée dans la liste **File type**.
	Compressed by	Cochez cette case et, dans la liste affiché, sélectionnez le type de compression du fichier source.
	Decrypt	Cochez cette case si le fichier est crypté du côté client à l'aide de Amazon S3. Pour plus d'informations, consultez Loading Encrypted Data Files from Amazon S3 (en anglais).
	Encryption key	Spécifiez la clé de cryptage utilisée pour crypter le fichier. Ce champ apparaît uniquement lorsque la case **Decrypt** est cochée.
	Encoding	Sélectionnez dans la liste le type d'encodage des données à charger.
	Date format	Sélectionnez l'un des éléments suivants afin de spécifier le format de date dans les données source : • **NONE** : aucun format de date n'est spécifié. • **PATTERN** : sélectionnez cet élément et spécifiez le format de date dans le champ affiché. Le format de date par défaut est YYYY-MM-DD. • **AUTO** : sélectionnez cet élément si vous souhaitez que Amazon Redshift reconnaisse et convertisse automatiquement le format de date.
	Time format	Sélectionnez l'un des éléments suivants dans la liste, afin de spécifier le format de l'heure dans les données source : • **NONE** : aucun format d'heure n'est spécifié. • **PATTERN** : sélectionnez cet élément et spécifiez le format de l'heure dans le champ affiché. Le format de l'heure par défaut est YYYY-MM-DD HH:MI:SS. • **AUTO** : sélectionnez cet élément si vous souhaitez qu'Amazon Redshift reconnaisse et convertisse le format de l'heure. • **EPOCHSECS** : sélectionnez cet élément si les données source sont représentées comme temps epoch, soit le nombre de secondes depuis Jan 1, 1970 00:00:00 UTC. • **EPOCHMILLISECS** : sélectionnez cet élément si les données source sont représentées comme temps epoch, soit le nombre de millisecondes depuis Jan 1, 1970 00:00:00 UTC.
	Settings	Cliquez sur le bouton **[+]** sous la table afin de spécifier plus de paramètres pour le chargement des données. • **Parameter** : cliquez sur la cellule et sélectionnez un paramètre dans la liste déroulante. • **Value** : configurez la valeur du paramètre correspondant. Notez que vous ne pouvez configurer la valeur d'un paramètre (par exemple **IGNOREBLANKLINES**) ne nécessitant pas de valeur. Pour plus d'informations concernant les paramètres, consultez http://docs.aws.amazon.com/redshift/latest/dg/r_COPY.html (en anglais).
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous	

	travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Le composant **tRedshiftBulkExec** supporte le chargement des données dans Amazon Redshift depuis un fichier délimité/CSV, JSON, ou à taille fixe, dans Amazon S3, mais le composant **tRedshiftOutputBulk** supporte uniquement la génération et le chargement d'un fichier délimité/CSV dans Amazon S3. Lorsque vous devez charger des données depuis un fichier JSON ou à taille fixe, vous pouvez utiliser le composant **tFileOutputJSON** ou **tFileOutputpositional** avec le **tS3Put** au lieu d'utiliser le composant **tRedshiftOutputBulk** pour générer et charger le fichier dans Amazon S3.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Charger/retirer des données dans/de Amazon S3

Ce scénario décrit un Job générant un fichier délimité, le chargeant le fichier dans S3, chargeant des données du fichier de S3 dans Redshift et les affichant dans la console. Ce Job retire ensuite les données de Redshift, les ajoute dans des fichiers S3 pour chaque slice du cluster Redshift puis liste et obtient les fichiers retirés dans S3.

Prérequis :

Les variables de contexte sont créées et sauvegardées dans la vue **Repository**. Pour plus d'informations concernant les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

- *redshift_host* : URL de l'endpoint de connexion au cluster Redshift.
- *redshift_port* : numéro du port d'écoute du serveur de la base de données.
- *redshift_database* : nom de la base de données.
- *redshift_username* : nom de l'utilisateur pour l'authentification à la base de données.
- *redshift_password* : mot de passe pour l'authentification à la base de données.
- *redshift_schema* : nom du schéma.
- *s3_accesskey* : clé d'accès à Amazon S3.
- *s3_secretkey* : clé secrète pour accéder à Amazon S3.
- *s3_bucket* : nom du bucket Amazon S3.

	Name	Type	Default Value
1	redshift_host	String	redshift-cluster.a1bcdefghij1.us-east-1.redshift.amazonaws.com
2	redshift_port	String	5439
3	redshift_database	String	redshiftdb
4	redshift_username	String	talend
5	redshift_password	Password	**********
6	redshift_schema	String	redshiftschema
7	s3_accesskey	String	ABCDEFGHIJKLMNOPQRST
8	s3_secretkey	String	aB1cDeF2gHiJk+l2mNoPQrsT+h6+6ypvDGTSRYU4
9	s3_bucket	String	redshift-bucket-talend

Notez que toutes les valeurs de contexte dans la capture d'écran ci-dessus ont pour seul objectif la démonstration.

Ajouter et relier les composants

1. Créez un nouveau Job et appliquez au Job toutes les variables de contexte listées ci-dessus.
2. Ajoutez les composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette** : un composant **tRowGenerator**, un **tRedshiftOutputBulk**, un **tRedshiftBulkExec**, un **tRedshiftInput**, un **tLogRow**, un **tRedshiftUnload**, un **tS3List** et un **tS3Get**.
3. Reliez le **tRowGenerator** au **tRedshiftOutputBulk** à l'aide d'un lien **Row** > **Main**.
4. Répétez l'opération afin de relier le **tRedshiftInput** au **tLogRow**.
5. Reliez le **tS3List** au **tS3Get** à l'aide d'un lien **Row** > **Iterate**.
6. Reliez le **tRowGenerator** au **tRedshiftBulkExec** à l'aide d'un lien **Trigger** > **On Subjob Ok**.
7. Répétez l'opération pour relier le **tRedshiftBulkExec** au **tRedshiftInput**, le **tRedshiftInput** au **tRedshiftUnload** et le **tRedshiftUnload** au **tS3List**.

Configurer les composants

Préparer un fichier et le charger dans S3

1. Double-cliquez sur le **tRowGenerator** afin d'ouvrir son éditeur **RowGenerator Editor**.

2. Cliquez deux fois sur le bouton **[+]** afin d'ajouter deux colonnes : *ID*, de type **Integer** et *Name*, de type **String**.

3. Cliquez dans la cellule de la colonne **Functions** et sélectionnez dans la liste une fonction pour chaque colonne. Dans cet exemple, sélectionnez **Numeric.sequence** pour générer des nombres en séquence pour la colonne *ID* et sélectionnez **TalendDataGenerator.getFirstName** afin de générer des prénoms aléatoires pour la colonne *Name*.

4. Dans le champ **Number of Rows for RowGenerator**, saisissez le nombre de lignes de données à générer. Dans cet exemple, saisissez *20*.

5. Cliquez sur **OK** pour fermer l'éditeur de schéma et acceptez la propagation proposée par la boîte de dialogue qui s'ouvre.

6. Double-cliquez sur le **tRedshiftOutputBulk** pour ouvrir sa vue **Basic settings**.

7. Dans le champ **Data file path at local**, spécifiez le chemin local pour le fichier à générer. Dans cet exemple, le chemin est *E:/Redshift/redshift_bulk.txt*.

8. Dans le champ **Access Key**, appuyez sur les touches **Ctrl + Espace** et, dans la liste, sélectionnez *context.s3_accesskey* afin de renseigner ce champ.

   Répétez l'opération dans le champ **Secret Key** avec la valeur *context.s3_accesskey* et dans le champ **Bucket** avec la valeur *context.s3_bucket*.

9. Dans le champ **Key**, saisissez un nouveau nom pour le fichier à générer après chargement dans Amazon S3. Dans cet exemple, saisissez *person_load*.

Charger les données du fichier de S3 dans Redshift

1. Double-cliquez sur le **tRedshiftBulkExec** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **Host**, appuyez sur les touches **Ctrl + Espace** et, dans la liste, sélectionnez *context.redshift_host* afin de renseigner ce champ.

   Répétez l'opération pour :

   - le champ **Port** avec la valeur *context.redshift_port*,
   - le champ **Database** avec la valeur *context.redshift_database*,
   - le champ **Schema** avec la valeur *context.redshift_schema*,
   - le champ **Username** avec la valeur *context.redshift_username*,
   - le champ **Password** avec la valeur *context.redshift_password*,
   - le champ **Access Key** avec la valeur *context.s3_accesskey*,
   - le champ **Secret Key** avec la valeur *context.s3_secretkey* et
   - le champ **Bucket** avec la valeur *context.s3_bucket*.

3. Dan le champ **Table Name**, saisissez le nom de la table à écrire. Dans cet exemple, saisissez *person*.

4. Dans la liste **Action on table**, sélectionnez **Drop table if exists and create**.

5. Dans le champ **Key**, saisissez le nom du fichier de Amazon S3 à charger. Dans cet exemple, saisissez *person_load*.

6. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et, dans la fenêtre qui s'ouvre, définissez le schéma en ajoutant deux colonnes : *ID* de type **Integer** et *Name*, de type **String**.

Récupérer les données de la table dans Redshift

1. Double-cliquez sur le composant **tRedshiftInput** pour ouvrir sa vue **Basic settings**.

2. Renseignez les champs **Host**, **Port**, **Database**, **Schema**, **Username** et **Password** avec les variables de contexte correspondantes.

3. Dans le champ **Table Name**, saisissez le nom de la table à lire. Dans cet exemple, saisissez *person*.

4. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et, dans la fenêtre qui s'ouvre, définissez le schéma en ajoutant deux colonnes : *ID* de type **Integer** et *Name*, de type **String**.

5. Dans le champ **Query**, saisissez l'instruction SQL suivante permettant de récupérer les données.

```
"SELECT * FROM" + context.redshift_schema + "person ORDER BY \"ID\""
```

6. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

7. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Retirer des données de Redshift et les ajouter dans des fichiers S3

1. Double-cliquez sur le **tRedshiftUnload** pour ouvrir sa vue **Basic settings**.

2. Renseignez les champs **Host**, **Port**, **Database**, **Schema**, **Username** et **Password** avec les variables de contexte correspondantes.

   Renseignez les champs **Access Key**, **Secret Key** et **Bucket** avec les variables de contexte correspondantes.

3. Dans le champ **Table Name**, saisissez le nom de la table de laquelle lire les données. Dans cet exemple, saisissez *person*.

4. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et, dans la fenêtre qui s'ouvre, définissez le schéma en ajoutant deux colonnes : *ID*, de type **Integer** et *Name*, de type **String**.

5. Dans le champ **Query**, saisissez l'instruction SQL suivante à partir de laquelle les résultats seront retirés.

```
"SELECT * FROM person"
```

6. Dans le champ **Key prefix**, saisissez le préfixe du nom pour les fichiers à retirer. Dans cet exemple, saisissez *person_unload_*.

Récupérer les fichiers retirés de Amazon S3

1. Double-cliquez sur le composant **tS3List** pour ouvrir sa vue **Basic settings**.

2. Renseignez les champs **Access Key** et **Secret Key** avec les variables de contexte correspondantes.

3. Dans la liste **Region**, sélectionnez la région AWS pour laquelle les fichiers retirés sont créés. Dans cet exemple, saisissez *US Standard*.

4. Décochez la case **List all buckets objects** et cliquez sur le bouton **[+]** sous la table pour ajouter une ligne.

   Renseignez la colonne **Bucket name** avec le nom du bucket dans lequel les fichiers retirés sont créés. Dans cet exemple, saisissez la variable de contexte *context.s3_bucket*.

   Renseignez la colonne **Key prefix** avec le préfixe du nom des fichiers retirés. Dans cet exemple, saisissez *person_unload_*.

5. Double-cliquez sur le **tS3Get** pour ouvrir sa vue **Basic settings**.

6. Renseignez les champs **Access Key** et **Secret Key** avec les variables de contexte correspondantes.

7. Dans la liste **Region**, sélectionnez la région AWS dans laquelle les fichiers retirés sont créés. Dans cet exemple, sélectionnez *US Standard*.

8. Dans le champ **Bucket**, saisissez le nom du bucket dans lequel les fichiers retirés sont créés. Dans cet exemple, saisissez la variable de contexte *context.s3_bucket*.

   Dans le champ **Key**, saisissez le nom des fichiers retirés, en appuyant sur les touches **Ctrl+Espace** et, dans la liste, sélectionnez la variable globale *((String)globalMap.get("tS3List_1_CURRENT_KEY"))*.

9. Dans le champ **File**, saisissez le chemin d'accès local à l'emplacement où les fichiers retirés sont sauvegardés. Dans cet exemple, saisissez *"E:/Redshift/" + ((String)globalMap.get("tS3List_1_CURRENT_KEY"))*.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder votre Job.

2. Exécutez le Job en appuyant sur **F6** ou en appuyant sur le bouton **Run** de la vue **Run**.

```
[statistics] connecting to socket on port 3726
[statistics] connected
.--+--------.
| tLogRow_1 |
|=-+-------=|
|ID|Name    |
|=-+-------=|
|1 |Abraham |
|2 |Herbert |
|3 |Chester |
|4 |John    |
|5 |Gerald  |
|6 |Lyndon  |
|7 |Warren  |
|8 |Calvin  |
|9 |Chester |
|10|Herbert |
|11|Ronald  |
|12|Herbert |
|13|Martin  |
|14|James   |
|15|Franklin|
|16|Herbert |
|17|Richard |
|18|Warren  |
|19|Jimmy   |
|20|George  |
'--+--------'
[statistics] disconnected
```

Comme affiché ci-dessus, les données générées sont écrites dans le fichier local *redshift_bulk.txt*. Le fichier est chargé dans S3 avec un nouveau nom, *person_load*. Les données sont chargées du fichier S3 dans la table *person* de Redshift et affichées dans la console. Les données sont ensuite retirées de la table *personn* de Redshift et ajoutées à deux fichiers *person_unload_0000_part_00* et *person_unload_0001_part_00* dans S3, par slice du cluster Redshift, puis les fichiers retirés de S3 sont listés et récupérés dans le dossier local.

tRedshiftClose

Propriétés du tRedshiftClose

Famille de composant	Databases/Amazon Redshift	
Fonction	Le composant **tRedshiftClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant est utilisé avec le **tRedShiftConnection** et le **tRedshiftCommit** pour assurer l'intégrité de la transaction effectuée dans la base de données.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tRedshiftConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec les composants Amazon Redshift, notamment le **tRedshiftConnection** et le **tRedshiftCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Manipuler des données avec Redshift*.

tRedshiftCommit

Propriétés du tRedshiftCommit

Famille de composant	Databases/Amazon Redshift	
Fonction	Le composant **tRedshiftCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tRedshiftConnection** dans la liste.
	Close Connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row >Main** pour relier le **tRedshiftCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Amazon Redshift, notamment le **tRedshiftConnection** et le **tRedshiftRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario du même type, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tRedshiftConnection

Propriétés du tRedshiftConnection

Famille de composant	Databases/Amazon Redshift	
Fonction	Le composant **tRedshiftConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus

		d’espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants Amazon Redshift, notamment les composants **tRedshiftCommit** et **tRedshiftRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Manipuler des données avec Redshift*.

tRedshiftInput

Propriétés du tRedshiftInput

Famille de composant	Databases/Amazon Redshift	
Fonction	Le composant **tRedshiftInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (**Main row**).	
Objectif	Le **tRedshiftInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de la base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table name	Nom de la table de base de données.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Guess schema	Cliquez sur ce bouton pour récupérer le schéma de la table.
Advanced settings	*Use cursor*	Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d'optimiser les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprime les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Amazon Redshift.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Manipuler des données avec Redshift

Ce scénario décrit un Job qui écrit des informations personnelles dans Redshift puis récupère ces informations et les affiche dans la console.

The scénario utilise les six composants suivants :

- un **tRedshiftConnection**, qui ouvre une connexion à Redshift.
- un **tFixedFlowInput**, qui définit la structure des informations personnelles et envoie ces données au composant suivant.
- un **tRedshiftOutput**, qui écrit les données reçues par le composant précédent dans Redshift.
- un **tRedshiftInput**, qui lit les données depuis Redshift.
- un **tLogRow**, qui affiche les données reçues par le composant précédent dans la console.

- et un **tRedshiftClose**, qui ferme la connexion à Redshift.

Déposer et relier les composants

1. Déposez les six composants listés précédemment depuis la **Palette** dans l'espace de modélisation graphique.
2. Reliez le **tFixedFlowInput** au **tRedshiftOutput** à l'aide d'un lien **Row** > **Main**.
3. Reliez le **tRedshiftInput** au **tLogRow**à l'aide d'un lien **Row** > **Main**.
4. Reliez le **tRedshiftConnection** au **tFixedFlowInput** à l'aide d'un lien **Trigger** > **OnSubjobOk**.
5. Reliez le **tFixedFlowInput** au **tRedshiftInput** et le **tRedshiftInput** au **tRedshiftClose** à l'aide de liens **Trigger** > **OnSubjobOk**.

Configurer les composants

Ouvrir une connexion à Redshift

1. Double-cliquez sur le **tRedshiftConnection** afin d'ouvrir sa vue **Basic settings**.

2. Sélectionnez **Built-In**depuis la liste **Property Type**.

   Dans les champs **Host**, **Port**, **Database**, **Schema**, **Username** et **Password**, saisissez les informations nécessaires pour se connecter à Redshift.

3. Dans la vue **Advanced settings**, cochez la case **Auto Commit** afin de commiter tous les changements à vers Redshift à chaque transaction.

Définir les données d'entrée

1. Double-cliquez sur le **tFixedFlowInput** afin d'ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir l'éditeur de schéma.

3. Dans l'éditeur de schéma, cliquez sur le bouton **[+]** pour ajouter trois colonnes : *id* de type **Integer**, *name* de type **String** et *age* de type **Integer**.

4. Cliquez sur **OK** pour valider les modifications puis acceptez la propagation proposée par la fenêtre pop-up **[Propagate]** en appuyant sur le bouton **Yes**.

5. Dans la zone **Mode**, sélectionnez **Use Inline Content (delimited file)** et saisissez les informations personnelles dans le champ **Content**.

```
1;Arthur;16
2;Ford;18
3;Jackson;17
```

Ecrire les données dans Redshift

1. Double-cliquez sur le composant **tRedshiftOutput** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** puis sélectionnez la connexion précédemment configurée dans le **tRedshiftConnection**, dans la liste déroulante **Component List**.

3. Dans le champ **Table**, parcourez ou saisissez le chemin d'accès à la table dans laquelle écrire les données, *redshiftexample* dans ce scénario.

4. Sélectionnez **Drop table if exists and create** dans la liste **Action on table** et sélectionnez **Insert** dans la liste **Action on data**.

5. Cliquez sur **Sync columns** afin de récupérer le schéma du composant précédent.

Récupérer les données de Redshift

1. Double-cliquez sur le **tRedshiftInput** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** et sélectionnez la connexion précédemment configurée dans le **tRedshiftConnection**, dans la liste **Component List**.

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur de schéma.

4. Dans l'éditeur de schéma, cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes, respectivement : *id*, de type **Integer**, *name*, de type **String** et *age*, de type **Integer**. La structure des données est la même que celle définie dans le **tFixedFlowInput**.

5. Cliquez sur **OK** afin de valider les modifications, puis acceptez la propagation proposée par la fenêtre **[Propagate]** qui s'ouvre.

6. Dans le champ **Table Name**, parcourez ou saisissez le chemin d'accès à la table dans laquelle vous souhaitez écrire les données, *redshiftexample* dans cet exemple.

7. Cliquez sur le bouton **Guess Query** pour générer la requête. Le champ **Query** contient automatiquement la requête générée.

Afficher les informations définies

1. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

2. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un meilleur affichage des résultats.

Fermer la connexion à Redshift

1. Double-cliquez sur le **tRedshiftClose** pour ouvrir sa vue **Basic settings**.

2. Dans la liste **Component List**, sélectionnez la connexion précédemment définie dans le **tRedshiftConnection**.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job.

2. Appuyez sur **F6** pour l'exécuter.

```
Starting job DOCT3191_tRedshift_S01 at 17:18 19/03/2014.

[statistics] connecting to socket on port 3808
[statistics] connected
.--+-------+---.
|  tLogRow_1   |
|=-+-------+--=|
|id|name   |age|
|=-+-------+--=|
|1 |Arthur |16 |
|2 |Ford   |18 |
|3 |Jackson|17 |
'--+-------+---'
[statistics] disconnected
Job DOCT3191_tRedshift_S01 ended at 17:18 19/03/2014. [exit code=0]
```

Les informations personnelles sont écrites dans la base de données cible Redshift et les données sont récupérées de la base de données et affichées dans la console.

tRedshiftOutput

Propriétés du tRedshiftOutput

Famille de composant	Databases/Amazon Redshift	
Fonction	Le composant **tRedshiftOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Objectif	Le **tRedshiftOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore.

		Create table if doesn't exist : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Extend Insert*	Cochez cette case pour effectuer une insertion de masse d'un ensemble de lignes défini et non les insérer ligne par ligne. Le gain de performance est considérable.

		Number of rows per insert : saisissez le nombre de lignes à insérer par opération. Notez que, plus la valeur spécifiée est grande, plus la performance est faible, à cause de l'augmentation de la sollicitation de la mémoire. Cette option est incompatible avec le lien **Reject**. Vous devez décochez la case si vous utilisez un lien **Row > Rejects** avec ce composant.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column: Saisissez une colonne de référence que le composant **tDBOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Amazon Redshift. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez *Scénario : Manipuler des données avec Redshift*.

tRedshiftOutputBulk

Propriétés du tRedshiftOutputBulk

Les composants **tRedshiftOutputBulk** et **tRedshiftBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus pour charger des données dans Amazon Redshift à partir d'un fichier CSV/délimité dans Amazon S3. Au cours de la première étape, un fichier délimlité/CSV est généré. Au cours de la seconde étape, le fichier est utilisé dans une instruction INSERT utilisée pour alimenter. Cette exécution en deux étapes est unifiée dans le composant **tRedshiftOutputBulkExec**. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans Amazon Redshift.

Famille de composant	Databases/Amazon Redshift	
Fonction	Le composant tRedshiftOutputBulk reçoit des données du composant précédent, génère un fichier délimité/CSV et charge le fichier dans Amazon S3.	
Objectif	Ce composant vous permet de préparer un fichier délimité/CSV pouvant être utilisé par le **tRedshiftBulkExec** pour alimenter Amazon Redshift.	
Basic settings	*Data file path at local*	Spécifiez le chemin d'accès local au fichier à générer. Notez que ce fichier est généré sur la machine où est installé le Studio ou sur laquelle le Job utilisant ce composant est déployé.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Append the local file	Cochez cette case pour écrire à la suite les données dans le fichier local spécifié s'il existe déjà, au lieu de l'écraser.
	Compress the data file	Cochez cette case et sélectionnez dans la liste un type de compression pour le fichier de données. Cette case est indisponible lorsque la case **Append the local file** est cochée.
S3 Setting	*Access Key*	Spécifiez l'ID de la clé d'accès identifiant de manière unique un compte AWS. Pour plus d'informations concernant l'obtention de votre clé d'accès et clé secrète, consultez Getting Your AWS Access Keys (en anglais).
	Secret Key	Spécifiez la clé secrète d'accès, constituant les informations de sécurité, ainsi que la clé d'accès.

		Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Bucket	Saisissez le nom du bucket Amazon S3 dans lequel charger le fichier.
	Key	Saisissez la clé d'objet assignée au fichier chargé dans Amazon S3.
Advanced settings	*Field Separator*	Saisissez le caractère utilisé pour séparer les champs.
	Text enclosure	Sélectionnez le caractère avec lequel entourer les champs.
	Delete local file after putting it to s3	Cochez cette case pour supprimer le fichier local après son chargement dans Amazon S3. Par défaut, cette case est cochée.
	Create directory if not exists	Cochez cette case pour créer le répertoire spécifié dans le champ **Data file path at local** s'il n'existe pas. Par défaut, cette case est cochée.
	Encoding	Sélectionnez un type d'encodage pour les données dans le fichier à générer.
S3 Setting	*Config client*	Cochez cette case afin de configurer les paramètres clients pour Amazon S3. Cliquez sur le bouton **[+]** sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre client et configurez les attributs suivants pour chaque paramètre : • **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante. • **Value** : saisissez la valeur du paramètre client correspondant.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tRedshiftBulkExec** pour alimenter Amazon Redshift avec un fichier délimité/CSV. Utilisés ensemble, ils permettent de maximiser les performances lorsqu'ils alimentent Amazon Redshift.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Charger/retirer des données dans/de Amazon S3*.

tRedshiftOutputBulkExec

Propriétés du tRedshiftOutputBulkExec

Les composants **tRedshiftOutputBulk** et **tRedshiftBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus pour charger des données dans Amazon Redshift à partir d'un fichier CSV/délimité dans Amazon S3. Au cours de la première étape, un fichier délimlité/CSV est généré. Au cours de la seconde étape, le fichier est utilisé dans une instruction INSERT utilisée pour alimenter. Cette exécution en deux étapes est unifiée dans le composant **tRedshiftOutputBulkExec**. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans Amazon Redshift.

Famille de composant	Databases/Amazon Redshift	
Fonction	Le composant **tRedshiftOutputBulkExec** reçoit des données du composant précédent, génère un fichier délimité/CSV, charge le fichier dans Amazon S3, puis charge les données d'Amazon S3 dans Redshift.	
Objectif	En tant que composant dédié, il permet un gain de performance pendant les opérations d'Insert dans Amazon Redshift.	
Basic settings	*Property Type*	Peut être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
Database settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Saisissez l'adresse IP ou le nom de l'hôte du serveur de la base de données.
	Port	Saisissez le numéro du port d'écoute du serveur de la base de données.
	Database	Saisissez le nom de la base de données.
	Schema	Saisissez le nom du schéma.
	Username et *Password*	Saisissez les données d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table Name	Spécifiez le nom de la table à écrire. Note que seule une table peut être écrite à la fois.
	Action on table	Sur la table définie, vous pouvez effectuer l'une des opérations suivantes : • **None** : n'effectuer aucune opération sur la table. • **Drop and create table** : supprimer la table puis en créer une nouvelle. • **Create table** : créer une table qui n'existe pas encore. • **Create table if not exists** : créer la table si nécessaire. • **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. • **Clear table** : supprimer le contenu de la table.

	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
File Generate Setting	*Data file path at local*	Spécifiez le chemin d'accès local du fichier à générer. Notez que le fichier est généré sur la machine où est installé le Studio ou sur celle où le Job utilisant ce composant est déployé.
	Append the local file	Cochez cette case pour écrire à la suite les données dans le fichier local spécifié s'il existe déjà, au lieu de l'écraser.
	Create directory if not exists	Cochez cette case pour créer le répertoire spécifié dans le champ **Data file path at local** s'il n'existe pas. Par défaut, cette case est cochée.
S3 Setting	*Access Key*	Spécifiez l'ID de la clé d'accès identifiant de manière unique un compte AWS. Pour plus d'informations concernant l'obtention de votre clé d'accès et clé secrète, consultez Getting Your AWS Access Keys (en anglais).
	Secret Key	Spécifiez la clé secrète d'accès, constituant les informations de sécurité, ainsi que la clé d'accès. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Bucket	Saisissez le nom du bucket Amazon S3 dans lequel charger le fichier.
	Key	Saisissez la clé d'objet assignée au fichier chargé dans Amazon S3.
Advanced settings	*Fields terminated by*	Saisissez le caractère utilisé pour sépare les champs.
	Enclosed by	Sélectionnez le caractère avec lequel entourer les champs.
	Compressed by	Cochez cette case et sélectionnez dans la liste un type de compression pour le fichier de données. Cette case est indisponible lorsque la case **Append the local file** est cochée.
	Encoding	Sélectionnez un type d'encodage pour les données dans le fichier à générer.
	Delete local file after putting it to s3	Cochez cette case pour supprimer le fichier local après son chargement dans Amazon S3. Par défaut, cette case est cochée.
	Date format	Sélectionnez l'un des éléments suivants afin de spécifier le format de date dans les données source : • **NONE** : aucun format de date n'est spécifié. • **PATTERN** : sélectionnez cet élément et spécifiez le format de date dans le champ affiché. Le format de date par défaut est YYYY-MM-DD. • **AUTO** : sélectionnez cet élément si vous souhaitez que Amazon Redshift reconnaisse et convertisse automatiquement le format de date.

	Time format	Sélectionnez l'un des éléments suivants dans la liste, afin de spécifier le format de l'heure dans les données source : • **NONE** : aucun format d'heure n'est spécifié. • **PATTERN** : sélectionnez cet élément et spécifiez le format de l'heure dans le champ affiché. Le format de l'heure par défaut est YYYY-MM-DD HH:MI:SS. • **AUTO** : sélectionnez cet élément si vous souhaitez qu'Amazon Redshift reconnaisse et convertisse le format de l'heure. • **EPOCHSECS** : sélectionnez cet élément si les données source sont représentées comme temps epoch, soit le nombre de secondes depuis Jan 1, 1970 00:00:00 UTC. • **EPOCHMILLISECS** : sélectionnez cet élément si les données source sont représentées comme temps epoch, soit le nombre de millisecondes depuis Jan 1, 1970 00:00:00 UTC.
	Settings	Cliquez sur le bouton **[+]** sous la table afin de spécifier plus de paramètres pour le chargement des données. • **Parameter** : cliquez sur la cellule et sélectionnez un paramètre dans la liste déroulante. • **Value** : configurez la valeur du paramètre correspondant. Notez que vous ne pouvez configurer la valeur d'un paramètre (par exemple **IGNOREBLANKLINES**) ne nécessitant pas de valeur. Pour plus d'informations concernant les paramètres, consultez http://docs.aws.amazon.com/redshift/latest/dg/r_COPY.html (en anglais).
S3 Setting	*Config client*	Cochez cette case afin de configurer les paramètres clients pour Amazon S3. Cliquez sur le bouton **[+]** sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre client et configurez les attributs suivants pour chaque paramètre : • **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante. • **Value** : saisissez la valeur du paramètre client correspondant.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est généralement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans Amazon Redshift.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé, consultez *Charger/retirer des données dans/de Amazon S3*.

tRedshiftRollback

Propriétés du tRedshiftRollback

Famille de composant	Databases/Amazon Redshift	
Fonction	Le **tRedshiftRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant permet d'éviter le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tRedshiftConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Amazon Redshift, notamment les composants **tRedshiftConnection** et **tRedshiftCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario du même type, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille*.

tRedshiftRow

Propriétés du tRedshiftRow

Famille de composant	Databases/Amazon Redshift	
Fonction	Le **tRedshiftRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tRedshiftRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table de base à lire.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**.	

	Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour des scénarios du même type, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données*.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL*.

tRedshiftUnload

Propriétés du tRedshiftUnload

Famille de composant	Databases/Amazon Redshift	
Fonction	Le composant **tRedshiftUnload** exécute une requête spécifiée dans Amazon Redshift et retire le résultat de la requête pour l'ajouter à un ou plusieurs fichiers dans Amazon S3.	
Objectif	Ce composant vous permet de retirer des données dans Amazon Redshift et les ajouter dans des fichiers dans Amazon S3.	
Basic settings	*Property Type*	Peut être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Propriété utilisée ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
Database settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Saisissez l'adresse IP ou le nom de l'hôte du serveur de la base de données.
	Port	Saisissez le numéro du port d'écoute du serveur de la base de données.
	Database	Saisissez le nom de la base de données.
	Schema	Saisissez le nom du schéma.
	Username et *Password*	Saisissez les données d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table Name	Saisissez le nom de la table de laquelle lire les données.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

	Query Type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
S3 Setting	*Access Key*	Spécifiez l'ID de la clé d'accès identifiant de manière unique un compte AWS. Pour plus d'informations concernant l'obtention de votre clé d'accès et clé secrète, consultez Getting Your AWS Access Keys (en anglais).
	Secret Key	Spécifiez la clé secrète d'accès, constituant les informations de sécurité, ainsi que la clé d'accès. Pour saisir la clé secrète, cliquez sur le bouton **[...]** à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Bucket	Saisissez le nom du bucket Amazon S3 duquel retirer les données.
	Key prefix	Saisissez le préfixe du nom pour les fichiers à retirer de Amazon S3. Par défaut, les fichiers retirés sont écrits par slice du cluster Redshift et les noms de fichiers sont écrits au format `<object_path>/<name_prefix><slice-number>_part_<file-number>`.
Advanced settings	*File type*	Sélectionnez le type des fichiers retirés de Amazon S3 dans la liste : • **Delimited file or CSV** : fichier délimité/CSV. • **Fixed width** : fichier à taille fixe.
	Fields terminated by	Saisissez le caractère utilisé pour séparer les champs. Ce champ est disponible uniquement lorsque l'option **Delimited file or CSV** est sélectionnée dans la liste **File type**.
	Enclosed by	Sélectionnez le caractère par lequel entourer les champs. Cette liste est disponible uniquement lorsque l'option **Delimited file or CSV** est sélectionnée dans la liste **File type**.
	Fixed width mapping	Saisissez une chaîne de caractères spécifiant un libellé et une largeur de colonne personnalisés entre guillemets doubles. Le format de la chaîne de caractères est : `ColumnLabel1:ColumnWidth1,ColumnLabel2:ColumnWidth2,....` Notez que le libellé de colonne dans la chaîne de caractères n'a aucune relation avec la nom de la colonne de la table et peut être une chaîne de caractères ou un entier. L'ordre des paires libellé/largeur doit correspondre à l'ordre des colonnes des tables. Ce champ est disponible uniquement lorsque l'option **Fixed width** est sélectionnée dans la liste **File type**.
	Compressed by	Cochez cette case et, dans la liste qui s'affiche, sélectionnez le type de compression des fichiers.
	Encrypt	Cochez cette case pour crypter les fichiers retirés du côté client à l'aide de Amazon S3. Pour plus d'informations, consultez Unloading Encrypted Data Files (en anglais).
	Encryption key	Saisissez la clé de cryptage utilisée pour crypter les fichiers retirés. Ce champ est disponible uniquement lorsque la case **Encrypt** est cochée.
	Specify null string	Cochez cette case et, dans la liste qui s'affiche, sélectionnez une chaîne de caractères représentant une valeur nulle dans les fichiers retirés.
	Escape	Cochez cette case pour placer un caractère d'échappement (\) avant chaque occurrence des caractères suivants pour les colonnes CHAR et VARCHAR dans les fichiers délimités retirés : interligne (\n), retour chariot (\r), le caractère délimiteur spécifié pour les données retirées, le caractère d'échappement (\), un guillemet (" ou ').
	Overwrite s3 object if exist	Cochez cette case pour écraser les fichiers d'objets existant dans Amazon S3.
	Parallel	Cochez cette case pour écrire les données en parallèle dans plusieurs fichiers retirés dans Amazon S3 selon le nombre de slices dans le cluster Redshift.

	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les requêtes SQL possibles pour la base de données Amazon Redshift.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Charger/retirer des données dans/de Amazon S3*.

tTeradataClose

Propriétés du tTeradataClose

Famille de composant	Databases/Teradata	
Fonction	Le composant **tTeradataClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Teradata.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tTeradataConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tTeradataConnection** et **tTeradataRollback**. Il est généralement utilisé avec un composant **tTeradataConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tTeradataClose**, consultez *tMysqlConnection*.

tTeradataCommit

Propriétés du tTeradataCommit

Famille de composant	Databases/Teradata	
Fonction	Le composant **tTeradataCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	Sélectionnez le composant **tTeradataConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tTeradataCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec des composants Teradata, notamment les composants **tTeradataConnection** et **tTeradataRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tTeradataCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tTeradataConnection

Propriétés du tTeradataConnection

Famille de composant	Databases/Teradata	
Fonction	Le composant **tTeradataConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Additional JDBC parameters	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple : CHARSET=KANJISIS_OS pour supporter les caractères japonais. Dans ce champ, vous pouvez paramétrer le type d'encodage.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Query band*	Cochez cette case pour utiliser la fonctionnalité Teradata Query Banding afin d'ajouter les métadonnées à la requête à traiter, comme le nom de l'utilisateur exécutant la requête. Cela peut vous permettre, par exemple, d'identifier l'origine de la requête. Une fois la case cochée, la table **Query Band parameters** s'affiche, dans laquelle vous pouvez saisir les informations des métadonnées à ajouter. Cette information prend la forme de paires clé/valeur, par

		exemple, *DpID* dans la colonne **Key** et *Finance* dans la colonne **Value**. Cette case permet de générer l'instruction SET QUERY_BAND FOR SESSION avec les paires clé/valeur déclarées dans la table **Query Band parameters**. Pour plus d'informations concernant cette instruction, consultez http://www.info.teradata.com/HTMLPubs/DB_TTU_14_00/index.html#page/SQL_Reference/B035_1144_111A/End_Logging-Syntax.027.143.html (en anglais).
	Auto Commit	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec des composants Teradata, notamment les composants **tTeradataCommit** et **tTeradataRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au composant **tTeradataConnection**, consultez *tMysqlConnection*.

tTeradataFastExport

Propriétés du tTeradataFastExport

Famille de composant	Databases/Teradata	
Fonction	Le composant **tTeradataFastExport** exporte très rapidement des lots de données volumineux d'une table ou d'une vue Teradata.	
Objectif	Le **tTeradataFastExport** exporte des lots de données d'une table Teradata vers un système client ou vers une base de données plus petite.	
Basic settings	*Use Commandline*	Cochez cette case pour activer le mode invite de commande.
	Use Java API	Cochez cette case pour activer le mode Java API.
	Property type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Execution platform	Sélectionnez le type de système d'exploitation que vous utilisez. Cette option est disponible en mode **Use Commandline**.
	Host	Adresse IP du serveur de base de données
	Database name	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Use query	Cochez cette case afin d'afficher le champ **Query** où saisir l'instruction SQL. Cette option est disponible en mode **Use Commandline**.
	Query	Saisissez l'instruction SQL dans le champ **Query**.
	Log database	Nom de la base de données de log. Cette option est disponible en mode **Use Commandline**.
	Log table	Nom de la table de log. Cette option est disponible en mode **Use Commandline**.
	Script generated folder	Parcourez votre répertoire et sélectionnez la destination du fichier qui sera créé. Cette option est disponible en mode **Use Commandline**.
	Exported file	Nom et chemin d'accès au fichier qui sera créé.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Row separator	Chaîne de caractères (par exemple : "\n"sous Unix) pour séparer des champs. Cette option est disponible en mode **Use Java API**.
	Error file	Parcourez votre répertoire et sélectionnez la destination du fichier dans lesquels seront enregistrés les messages d'erreur. Cette option est disponible en mode **Use Java API**.
Advanced settings	*Output*	Deux options sont disponibles dans la liste : **Output error to file** : écrit en sortie l'erreur dans le fichier spécifié dans le champ **Error log** et continue le traitement. **Output error to console** : écrit en sortie l'erreur dans la console et arrête le traitement. Disponible en mode **Use Commandline**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Teradata.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Si vous avez sélectionné le mode **Use Commandline**, vous devez installer le client Teradata sur la machine où se trouve le Job comprenant un **tTeradataFastExport**.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tTeradataFastLoad

Propriétés du tTeradataFastLoad

Famille de composant	Databases/Teradata	
Fonction	Le composant **tTeradataFastLoad** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tTeradataFastLoad** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Execute Batch every	Nombre de lignes par lot à charger.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Teradata.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tTeradataFastLoadUtility

Propriétés du tTeradataFastLoadUtility

Famille de composant	Databases/Teradata	
Fonction	Le composant **tTeradataFastLoadUtility** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tTeradataFastLoadUtility** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (**Main row**).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Execution platform	Sélectionnez le type de système d'exploitation que vous utilisez.
	Host	Nom de l'hôte ou l'adresse IP du serveur de la base de données.
	Database name	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant,

		sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Script generated folder	Parcourez votre répertoire et sélectionnez la destination du fichier qui sera créé.
	Load file	Entrez le nom du fichier à partir duquel vous souhaitez charger les données.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Error file	Parcourez votre répertoire et sélectionnez la destination du fichier dans lesquels seront enregistrés les messages d'erreur.
Advanced settings	*Define character set*	Spécifiez l'encodage des caractères dont vous avez besoin pour utiliser votre système.
	Check point	Saisissez la valeur du point de validation.
	Error files	Saisissez le nom du fichier dans lequel les messages d'erreur sont stockés. Par défaut, le code saisi est `ERRORFILES table_ERR1`, ce qui signifie que les deux tables *table_ERR1* et *table_ERR2* sont utilisées pour enregistrer les messages d'erreur.
	Return fastload error	Cochez cette case pour spécifier le code de retour à partir duquel retourner une erreur dans la console.
	ERRLIMIT	Saisissez le nombre limite d'erreurs détectées durant la phase de chargement. Le processus s'arrête lorsque la limite est atteinte. Par défaut, la valeur limite d'erreurs est de 1000000. Pour plus d'informations, consultez la documentation *Teradata FastLoad Reference*.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Teradata.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Insérer des données dans une table d'une base de données Teradata*.

tTeradataInput

Propriétés du tTeradataInput

Famille de composant	Databases/ Teradata	
Fonction	Le composant **tTeradataInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tTeradataInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple : CHARSET=KANJISIS_OS pour supporter les caractères japonais. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	Query band	Cochez cette case pour utiliser la fonctionnalité Teradata Query Banding afin d'ajouter les métadonnées à la requête à traiter, comme le nom de l'utilisateur exécutant la requête. Cela peut vous permettre, par exemple, d'identifier l'origine de la requête. Une fois la case cochée, la table **Query Band parameters** s'affiche, dans laquelle vous pouvez saisir les informations des métadonnées à ajouter. Cette information prend la forme de paires clé/valeur, par exemple, *DpID* dans la colonne **Key** et *Finance* dans la colonne **Value**. Cette case permet de générer l'instruction SET QUERY_BAND FOR SESSION avec les paires clé/valeur déclarées dans la table **Query Band parameters**. Pour plus d'informations concernant cette instruction, consultez http://www.info.teradata.com/HTMLPubs/DB_TTU_14_00/index.html#page/SQL_Reference/B035_1144_111A/End_Logging-Syntax.027.143.html (en anglais). Cette case est indisponible lorsque vous avez coché la case **Using an existing connection**. Dans ce cas, si vous devez utiliser la fonctionnalité **Query band**, configurez-la dans l'onglet **Advanced settings** du composant de connexion Teradata à utiliser.
	tStat Catcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé, consultez :

- *Scénario : Charger des données dans une base de données Teradata*.
- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.
- *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement*.
-

tTeradataMultiLoad

Propriétés du tTeradataMultiLoad

Famille de composant	Databases/Teradata	
Fonction	Le composant **tTeradataMultiLoad** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tTeradataMultiLoad** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (**Main row**).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Execution platform	Sélectionnez le type de système d'exploitation que vous utilisez.
	Host	Nom de l'hôte ou l'adresse IP du serveur de la base de données.
	Database name	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant,

		sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Script generated folder	Parcourez votre répertoire et sélectionnez la destination du fichier qui sera créé.
	Action to data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations Update et Delete. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (**Advanced settings**) pour une utilisation optimisée de ces opérations.*
	Where condition in case Delete	Saisissez une condition, qui, si elle est vérifiée, supprimera la ligne. Ce champ apparaît uniquement lorsque Delete est sélectionné dans la liste déroulante **Action to data**.
	Load file	Entrez le nom du fichier à partir duquel vous souhaitez charger les données.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Error file	Parcourez votre répertoire et sélectionnez la destination du fichier dans lesquels seront enregistrés les messages d'erreur.
Advanced settings	*Define Log table*	Cochez cette case pour définir la table de log que vous souhaitez utiliser au lieu de celle par défaut, à savoir celle que vous avez définie dans l'onglet **Basic settings**. La syntaxe requise pour définir la table de log est *databasename.logtablename.*
	BEGIN LOAD	Ce champ vous permet de définir votre commande BEGIN LOAD pour lancer ou relancer une tâche TPump. Vous pouvez spécifiez le nombre de sessions à utiliser, la limite d'erreurs, ainsi que tout autre paramètre nécessaire à l'exécution de la tâche. Pour plus d'informations, consultez la documentation *Teradata MultiLoad Reference*.
	Return mload error	Cochez cette case pour spécifier le code de retour à partir duquel retourner une erreur dans la console.
	Define character set	Spécifiez l'encodage des caractères dont vous avez besoin pour utiliser votre système.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**EXIT_VALUE** : code de sortie du processus. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Teradata.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé, consultez *Scénario : Insérer des données dans une table d'une base de données Teradata*.

tTeradataOutput

Propriétés du tTeradataOutput

Famille de composant	Databases/ Teradata	
Fonction	Le composant **tTeradataOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	**Le tTeradataOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	La liste **Action on table** est indisponible si vous cochez la case **Enable parallel execution**dans la vue **Advanced settings**. Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore.

		Create table if doesn't exist : créer la table si nécessaire. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Create	Elle n'est pas visible par défaut, sauf si vous choisissez de créer une table à partir de la liste déroulante **Action on table**. La table à créer peut être : -SET TABLE: table ne permettant pas de dupliquer les lignes. -MULTI SET TABLE : table permettant de dupliquer les lignes.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.

Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple : CHARSET=KANJISIS_OS pour supporter les caractères japonais. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution. Cette option n'est pas disponible si vous avez coché la case **Use an existing connection** dans la vue **Basic settings**. Si vous avez sélectionné **Drop and create table**, **Create table**, **Create table if does not exist** ou **Drop table if exists and create** dans la liste **Action on table**, dans la vue **Basic settings**, saisissez *0* dans le champ afin d'assurer la validité des instructions SQL. Pour plus d'informations concernant la validité des instructions SQL dans une base de données Teradata, consultez info.teradata.com (en anglais).
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tTeradataOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Query band	Cochez cette case pour utiliser la fonctionnalité Teradata Query Banding afin d'ajouter les métadonnées à la requête à traiter, comme le nom de l'utilisateur exécutant la requête. Cela peut vous permettre, par exemple, d'identifier l'origine de la requête. Une fois la case cochée, la table **Query Band parameters** s'affiche, dans laquelle vous pouvez saisir les informations des métadonnées à ajouter. Cette information prend la forme de paires clé/valeur, par exemple, *DpID* dans la colonne **Key** et *Finance* dans la colonne **Value**. Cette case permet de générer l'instruction SET QUERY_BAND FOR SESSION avec les paires clé/valeur déclarées dans la table **Query Band parameters**. Pour plus d'informations concernant cette instruction, consultez http://www.info.teradata.com/HTMLPubs/DB_TTU_14_00/index.html#page/SQL_Reference/B035_1144_111A/End_Logging-Syntax.027.143.html (en anglais). Cette case est indisponible lorsque vous avez coché la case **Using an existing connection**. Dans ce cas, si vous devez utiliser la fonctionnalité **Query band**, configurez-la dans l'onglet **Advanced settings** du composant de connexion Teradata à utiliser.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Use Batch	Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.
	Batch Size	Spécifiez le nombre d'enregistrements à traiter dans chaque lot. La valeur par défaut de ce champ est *10000*. Ce champ est disponible uniquement lorsque la case **Use Batch** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans	

	un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Teradata. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tTeradataRollback

Propriétés du tTeradataRollback

Famille de composant	Databases/Teradata	
Fonction	Le **tTeradataRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tTeradataConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Teradata, notamment les composants **tTeradataConnection** et **tTeradataCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tTeradataRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille*.

tTeradataRow

Propriétés du tTeradataRow

Famille de composant	Databases/ Teradata	
Fonction	Le **tTeradataRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tTeradataRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à traiter.
	Query type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.
	Query	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple : CHARSET=KANJISIS_OS pour supporter les caractères japonais. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Query band	Cochez cette case pour utiliser la fonctionnalité Teradata Query Banding afin d'ajouter les métadonnées à la requête à traiter, comme le nom de l'utilisateur exécutant la requête. Cela peut vous permettre, par exemple, d'identifier l'origine de la requête. Une fois la case cochée, la table **Query Band parameters** s'affiche, dans laquelle vous pouvez saisir les informations des métadonnées à ajouter. Cette information prend la forme de paires clé/valeur, par exemple, *DpID* dans la colonne **Key** et *Finance* dans la colonne **Value**. Cette case permet de générer l'instruction SET QUERY_BAND FOR SESSION avec les paires clé/valeur déclarées dans la table **Query Band parameters**. Pour plus d'informations concernant cette instruction, consultez http://www.info.teradata.com/HTMLPubs/DB_TTU_14_00/index.html#page/SQL_Reference/B035_1144_111A/End_Logging-Syntax.027.143.html (en anglais). Cette case est indisponible lorsque vous avez coché la case **Using an existing connection**. Dans ce cas, si vous devez utiliser la fonctionnalité **Query band**, configurez-la dans l'onglet **Advanced settings** du composant de connexion Teradata à utiliser.

	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez :

- *Scénario : Charger des données dans une base de données Teradata*.
- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tTeradataTPTExec

Propriétés du tTeradataTPTExec

Famille de composant	Databases/Teradata	
Fonction	Le composant **tTeradataTPTExec** est une combinaison du **tTeradataFastLoad**, du **tTeradataMultiLoad**, du **tTeradataTPump** et du **tTeradataFastExport**. Il charge les données d'un fichier existant dans une base de données Teradata.	
Objectif	Le **tTeradataTPTExec** offre de hautes performances lors de l'insertion des données d'un fichier existant dans une table d'une base de données Teradata.	
Basic settings	*Property Type*	Peut être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants concernant la connexion à la base de données sont alors pré-remplis à l'aide des données collectées.
	Execution platform	Sélectionnez le type de système d'exploitation utilisé pour exécuter le Job, **Windows** ou **Unix**.
	TDPID	Spécifiez le Director program identifier de Teradata. Cela peut-être le nom ou l'adresse IP de la base de données Teradata à laquelle vous accédez.
	Database name	Saisissez dans ce champ le nom de la base de données Teradata.
	Username et *Password*	Spécifiez l'identifiant et mot de passe de l'utilisateur de la base de données Teradata. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Il est recommandé de ne pas utiliser le mot-clé Teradata Database comme nom de colonne de base de données. Si vous y êtes obligé, le nom de la colonne de base de données doit alors être entouré par \". Par exemple, lorsque le mot-clé *id* est utilisé comme nom de colonne d'une base de données, le champ **Db Column** doit être renseigné par *\"id \"*.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Consumer Operator	Sélectionnez un opérateur dans la liste déroulante. • **Load** : charge un grand volume de données dans une table vide dans la base de données Teradata à l'aide de l'utilitaire Teradata FastLoad utility. • **Inserter** : insère les données dans une table définie dans la base de données Teradata. • **Update** : réalise des insertions, des mises à jour ou des suppressions dans une table nouvellement créée ou déjà existante de la base de données Teradata à l'aide de l'utilitaire Teradata MultiLoad utility. • **Stream** : réalise des insertions, des mises à jour ou des suppressions dans une table nouvellement créée ou déjà existante de la base de données Teradata à l'aide de l'utilitaire Teradata TPump. Pour plus d'informations concernant cet opérateur, consultez http://developer.teradata.com/sites/all/files/documentation/linked_docs/html/online/B035-2436-088A/wwhelp/wwhimpl/js/html/wwhelp.htm (en anglais).
	Action on data	Sélectionnez dans la liste déroulante une action à effectuer sur les données. • **Insert** : ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. • **Update** : mettre à jour des entrées déjà existantes dans la table. • **InsertOrUpdate** : ajouter de nouvelles entrées. Une mise à jour sera effectuée si l'entrée avec la référence donnée existe déjà. • **Delete** : Supprimer des entrées correspondant au flux d'entrée. Ce champ apparaît uniquement lorsque l'opérateur **Update** ou **Stream** est sélectionné dans la liste déroulante **Consumer Operator**.
	Producer Operator	Sélectionnez un opérateur dans la liste déroulante. Seul l'opérateur **DataConnector** est actuellement supporté. Pour plus d'informations concernant cet opérateur, consultez http://developer.teradata.com/sites/all/files/documentation/linked_docs/html/online/B035-2436-088A/wwhelp/wwhimpl/js/html/wwhelp.htm (en anglais).
	Table	Spécifiez le nom de la table à écrire dans la base de données Teradata. Notez que vous ne pouvez écrire qu'une table à la fois.
	Script generated folder	Spécifiez le répertoire sous lequel le fichier script Teradata Parallel Transporter sera créé durant l'exécution du Job puis supprimé avant la fin du Job.
	Load file	Spécifiez le fichier contenant les données à charger dans la base de données Teradata.
	Error file	Spécifiez le fichier dans lequel les messages de log seront enregistrés.

Advanced settings	*Field separator*	Caractère, chaîne ou expression régulière séparant les champs.
	Define Log table	Cochez cette case pour spécifiez le nom de table de log.
	Set Script Parameters	Cochez cette case pour spécifier la valeur des paramètres **Load Operator**, **Data Connector**, **Job Name**, and **Layout Name (schema)** qui seront utilisés lors de la génération d'un script durant l'exécution du Job. Si vous ne les spécifiez pas manuellement, le système utilisera leurs valeurs par défaut.
	Load Operator	Spécifiez l'opérateur de chargement. Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée.
	Data Connector	Spécifiez le connecteur de données. Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée.
	Job Name	Spécifiez le nom du Job TPT (Teradata Parallel Transporter) défini via la commande Teradata *tbuild*. Pour plus d'informations concernant la commande *tbuild*, consultez http://developer.teradata.com/sites/all/files/documentation/linked_docs/html/online/B035-2436-088A/wwhelp/wwhimpl/js/html/wwhelp.htm (en anglais). Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée.
	Layout Name(schema)	Spécifiez un schéma pour le chargement des données. Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée.
	Return mload error	Cochez cette case pour spécifier le code de retour à partir duquel retourner une erreur dans la console.
	Define character set	Cochez cette case pour spécifier l'encodage des caractères à utiliser dans votre système.
	Apply TPT consumer operator optional attributes	Cochez cette case afin de définir les attributs facultatifs pour l'opérateur du consommateur sélectionné. Pour plus d'informations concernant les attributs facultatifs de chaque opérateur de consommateur, consultez le Guide *Teradata Parallel Transporter Reference* (en anglais) à l'adresse suivante : http://developer.teradata.com/sites/all/files/documentation/linked_docs/2436020A_TPT-Reference-13.10.pdf.
	Optional attributes	Cliquez sur le bouton **[+]** sous la table pour ajouter autant des lignes que nécessaire pour chaque attribut facultatif. Configurez les paramètres comme suit pour chaque attribut : • **Name** : cliquez dans la cellule et sélectionnez un attribut facultatif dans la liste déroulante. La liste des attributs facultatifs varie selon l'opérateur du consommateur sélectionné. Pour plus d'informations concernant les attributs facultatifs pour chaque opérateur du consommateur, consultez *Attributs facultatifs supportés pour chaque opérateur de consommateur*. • **Value** : saisissez la valeur de l'attribut facultatif. Pour plus d'informations concernant la définition de la valeur d'un attribut, consultez le Guide *Teradata Parallel Transporter Reference* (en anglais) à l'adresse suivante : http://developer.teradata.com/sites/all/files/documentation/linked_docs/2436020A_TPT-Reference-13.10.pdf. Cette table apparaît uniquement lorsque la case **Apply TPT consumer operator optional attributes** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables	**EXIT_VALUE** : code de sortie du processus. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut s'utiliser en standalone dans un Job ou un sous-job. Il offre de hautes performances lors de l'insertion de données d'un fichier existant dans une table d'une base de données Teradata. Pour plus d'informations concernant l'utilisation de ce composant, consultez http://developer.teradata.com/sites/all/files/documentation/linked_docs/html/online/B035-2436-088A/wwhelp/wwhimpl/js/html/wwhelp.htm (en anglais).
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Il est nécessaire d'installer l'outil de client Teradata avec les fonctionnalités Teradata Parallel Transporter Base et Teradata Parallel Transporter Stream sur la machine où les Jobs utilisant ce composant sont exécutés.

Attributs facultatifs supportés pour chaque opérateur de consommateur

Cette section liste tous les attributs facultatifs supportés pour chaque opérateur de consommateur.

- Pour l'opérateur **Load** :

  BufferSize, ErrorLimit, MaxSessions, MinSessions, TenacityHours, TenacitySleep, AccountId, DataEncryption, DateForm, ErrorTable1, ErrorTable2, LogonMech, LogonMechData, NotifyExit, NotifyExitlsDLL, NotifyLevel, NotifyMethod, LogSQL, NotifyString, PauseAcq, PrivateLogName, QueryBandSessInfo, WildcardInsert, WorkingDatabase, TraceLevel.

- Pour l'opérateur **Inserter** :

  AccountId, DataEncryption, DateForm, LogonMech, LogonMechData, LogSQL, PrivateLogName, QueryBandSessInfo, ReplicationOverride, TraceLevel, WorkingDatabase.

- Pour l'opérateur **Update** :

  PrivateLogName, BufferSize, ErrorLimit, MaxSessions, MinSessions, TenacityHours, TenacitySleep, AccountId, AmpCheck, DataEncryption, DateForm, DeleteTask, DropErrorTable, DropLogTable, DropWorkTable, ErrorTable1, ErrorTable2, LogonMech, LogonMechData, LogSQL, NotifyExit, NotifyExitIsDLL, NotifyLevel, NotifyMethod, NotifyString, PauseAcq, QueryBandSessInfo, QueueErrorTable, WorkingDatabase, WorkTable, TraceLevel.

- Pour l'opérateur **Stream** :

  Buffers, ErrorLimit, MaxSessions, MinSessions, Pack, Rate, Periodicity, TenacityHours, TenacitySleep, AccountId, AppendErrorTable, ArraySupport, DataEncryption, DateForm, DropErrorTable, DropMacro, ErrorTable, LogonMech, LogonMechData, MacroDatabase, OperatorCommandID, NotifyExit, NotifyExitlsDLL, NotifyLevel, NotifyMethod, LogSQL, NotifyString, PackMaximum, PrivateLogName, QueryBandSessInfo, QueueErrorTable, ReplicationOverrride, Robust, WorkingDatabase, TraceLevel.

Scénario : Charger des données dans une base de données Teradata

Ce scénario décrit un Job créant une table dans une base de données Teradata, écrivant des données dans un fichier délimité, chargeant ces données du fichier dans la table, récupérant les données de la table et les affichant dans la console.

Construire le Job

1. Créez un Job et ajoutez les composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou bien en les glissant-déposant de la **Palette** dans l'espace de modélisation graphique : un **tTeradataRow**, un **tFixedFlowInput**, un **tFileOutputDelimited**, un **TeradataTPTExec**, un **tTeradataInput** et un **tLogRow**.

2. Reliez le **tFixedFlowInput** au **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main**.

3. Connectez le **tTeradataInput** au **tLogRow** avec le même type de lien.

4. Reliez le **tTeradataRow** au **tFixedFlowInput** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

5. Connectez le **tFixedFlowInput** au **TeradataTPTExec** et le **TeradataTPTExec** au **tTeradataInput** à l'aide du même type de lien.

Configurer les composants

Créer une nouvelle table d'une base de données Teradata

1. Double-cliquez sur le **tTeradataRow** pour ouvrir sa vue **Basic settings**.

2. Renseignez les champs **Host**, **Database**, **Username** et **Password** en saisissant respectivement le nom de l'hôte, le nom de la base de données, l'identifiant et le mot de passe de l'utilisateur, pour la connexion.

3. Dans le champ **Query**, saisissez l'instruction SQL suivante afin de créer une table nommée *person* contenant ces trois colonnes :*id*, *name*, *sex*.

```
CREATE SET TABLE samples.person,
FALLBACK,
NO BEFORE JOURNAL,
NO AFTER JOURNAL
(
  id INTEGER NOT NULL,
  name VARCHAR(50),
  sex VARCHAR(20)
)
UNIQUE PRIMARY INDEX (id)
```

Préparer les données source

1. Double-cliquez sur le composant **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

3. Cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes : *id* de **Type Integer**, *name* et *sex*, de **Type String**.

4. Cliquez sur **OK** pour fermer l'éditeur et acceptez la propagation lorsqu'une fenêtre vous propose de propager le schéma.

5. Dans la zone **Mode**, sélectionnez **Use Inline Content (delimited file)** et saisissez les données d'entrée dans le champ **Content**.

```
1;Ford;Male
2;Rose;Female
3;Sabrina;Female
4;Teddy;Male
5;Kate;Male
```

6. Double-cliquez sur le **tFileOutputDelimited** pour ouvrir sa vue **Basic settings**.

7. Dans le champ **File Name**, spécifiez le fichier dans lequel écrire les données d'entrée. Dans cet exemple, saisissez *E:/person.csv*.

Charger les données sources dans une table vide

1. Double-cliquez sur le **TeradataTPTExec** pour ouvrir sa vue **Basic settings**.

2. Dans les champs **TDPID**, **Database name**, **Username** et **Password**, saisissez vos informations de connexion à la base de données Teradata.

3. Dans le champ **Table**, saisissez le nom de la table dans laquelle les données source seront chargées. Dans cet exemple, saisissez *person*.

4. Dans le champ **Script generated folder**, parcourez votre système jusqu'au répertoire sous lequel créer le fichier du script Teradata Parallel Transporter durant l'exécution du Job. Dans cet exemple, le chemin est *E:/*.

5. Dans le champ **Load file**, parcourez votre système jusqu'au fichier contenant les données source. Dans cet exemple, le chemin est *E:/person.csv*.

6. Dans le champ **Error file**, spécifiez le fichier dans lequel enregistrer les messages de log. Dans cet exemple, le chemin est *E:/error.log*.

7. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** to open the schema editor.

Cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes : *id* de **Type Integer**, *name* et *sex*, de **Type String**. Le nom des colonnes *id* et *name* dans la colonne **Db Column** sont entourés par \", car ce sont des mots-clés de la base de données Teradata.

Cliquez sur **OK** pour valider ces modifications et fermer l'éditeur du schéma.

8. Cliquez sur l'onglet **Advanced settings** pour ouvrir sa vue. Cochez la case **Apply TPT consumer operator optional attributes** et cliquez quatre fois sur le bouton **[+]** sous la table **Optional attributes** pour ajouter les attributs nécessaires suivants : **ErrorLimit**, **ErrorTable1**, **QueryBandSessInfo** et **TraceLevel**.

Pour les attributs de type VARCHAR, saisissez les valeurs entre guillemets doubles.

Récupérez les données de la table de base de données Teradata

1. Double-cliquez sur le **tTeradataInput** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **Table Name**, saisissez le nom de la table de laquelle lire les données. Dans cet exemple, saisissez *person*.

3. Dans le champ **Query**, saisissez l'instruction SQL suivante pour récupérer les données de la table *person*.

```
SELECT * FROM samples.person ORDER BY id
```

4. Cliquez sur le **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

5. Cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes : *id* de **Type Integer**, *name* et *sex*, de **Type String**. Le nom des colonnes *id* et *name* dans la colonne **Db Column** sont entourés par \", car ce sont des mots-clés de la base de données Teradata.

6. Cliquez sur **OK** pour fermer l'éditeur de schéma et acceptez la propagation proposée par la fenêtre qui s'ouvre.

7. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

8. Dans la zone **Mode**, sélectionnez l'option **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pou sauvegarder le Job.

2. Appuyez sur **F6** pour l'exécuter.

```
Starting job tTeradataTPTExec_Demo at 10:35 28/09/2014.

[statistics] connecting to socket on port 3553
[statistics] connected
.--+-------+------.
|     tLogRow_1    |
|=-+-------+-----=|
|id|name   |sex   |
|=-+-------+-----=|
|1 |Ford   |Male  |
|2 |Rose   |Female|
|3 |Sabrina|Female|
|4 |Teddy  |Male  |
|5 |Kate   |Male  |
'--+-------+------'

[statistics] disconnected
Job tTeradataTPTExec_Demo ended at 10:35 28/09/2014. [exit code=0]
```

Les données écrites dans la table de la bases de données Teradata spécifiée sont affichées dans la console.

tTeradataTPTUtility

Propriétés du tTeradataTPTUtility

Famille de composant	Databases/Teradata	
Fonction	Le composant **tTeradataTPTUtility** combine les utilitaires des composants **tTeradataFastLoad**, **tTeradataMultiLoad**, **tTeradataTPump** et **tTeradataFastExport** et permet d'écrire des données entrantes dans un fichier puis de charger les données du fichier dans une base de données Teradata.	
Objectif	Le **tTeradataTPTUtility** permet d'écrire les données entrantes dans un fichier puis de charger les données du fichier dans une base de données Teradata.	
Basic settings *Data file*	*Property Type*	Peut être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Le champ **Filename** est alors pré-rempli à l'aide des données collectées.
	Filename	Spécifiez le fichier dans lequel sauvegarder vos données de sortie.
	Append	Cochez cette case pour ajouter les données entrantes dans le fichier défini dans le champ **Filename**.
Connection	*Property Type*	Peut-être **Built-In** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants concernant la connexion à la base de données sont alors pré-remplis à l'aide des données collectées.
	Execution platform	Sélectionnez le type de système d'exploitation utilisé pour exécuter le Job, **Windows** ou **Unix**.
	TDPID	Spécifiez le Director program identifier de Teradata. Cela peut-être le nom ou l'adresse IP de la base de données Teradata à laquelle vous accédez.
	Database name	Saisissez dans ce champ le nom de la base de données Teradata.
	Username et *Password*	Spécifiez l'identifiant et mot de passe de l'utilisateur de la base de données Teradata. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Il est recommandé de ne pas utiliser le mot-clé Teradata Database comme nom de colonne de base de données. Si vous y êtes obligé, le nom de la colonne de base de données doit alors être entouré par \". Par exemple, lorsque le mot-clé

		id est utilisé comme nom de colonne d'une base de données, le champ **Db Column** doit être renseigné par *\"id\"*.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Consumer Operator	Sélectionnez un opérateur dans la liste déroulante. • **Load** : Charge un grand volume de données dans une table vide dans la base de données Teradata à l'aide de l'utilitaire Teradata FastLoad utility. • **Inserter** : Insère les données dans une table définie dans la base de données Teradata. • **Update** : Réalise des insertions, des mises à jour ou des suppressions dans une table nouvellement créée ou déjà existante de la base de données Teradata à l'aide de l'utilitaire Teradata MultiLoad utility. • **Stream** : Réalise des insertions, des mises à jour ou des suppressions dans une table nouvellement créée ou déjà existante de la base de données Teradata à l'aide de l'utilitaire Teradata TPump. Pour plus d'informations concernant cet opérateur, consultez http://developer.teradata.com/sites/all/files/documentation/linked_docs/html/online/B035-2436-088A/wwhelp/wwhimpl/js/html/wwhelp.htm (en anglais).
	Action on data	Sélectionnez dans la liste déroulante une action à effectuer sur les données. • **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. • **Update** : Mettre à jour des entrées déjà existantes dans la table. • **InsertOrUpdate** : Ajouter de nouvelles entrées. Une mise à jour sera effectuée si l'entrée avec la référence donnée existe déjà. • **Delete** : Supprimer des entrées correspondant au flux d'entrée. Vous devez spécifier au moins une colonne comme clé primaire sur laquelle baser l'opération **Update** ou **Delete**. Cela est faisable en cliquant sur le bouton **[...]** à côté du champ **Edit schema** et en cochant la case de la (des) colonne(s) à définir comme clé(s) primaire(s). Ce champ apparaît uniquement lorsque l'opérateur **Update** ou **Stream** est sélectionné dans la liste déroulante **Consumer Operator**.
	Producer Operator	Sélectionnez un opérateur dans la liste déroulante. Seul l'opérateur **DataConnector** est actuellement supporté.

		DataConnector : accède aux fichiers directement ou via un module d'accès, puis les écrit dans le flux de données. Pour plus d'informations concernant cet opérateur, consultez http://developer.teradata.com/sites/all/files/documentation/linked_docs/html/online/B035-2436-088A/wwhelp/wwhimpl/js/html/wwhelp.htm (en anglais).
	Table	Spécifiez le nom de la table à écrire dans la base de données Teradata. Notez que vous ne pouvez écrire qu'une table à la fois.
	Script generated folder	Spécifiez le répertoire sous lequel le fichier script Teradata Parallel Transporter sera créé durant l'exécution du Job puis supprimé avant la fin du Job.
	Error file	Spécifiez le fichier dans lequel les messages de log seront enregistrés.
Advanced settings	*Row separator*	Caractère, chaîne ou expression régulière séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Set Script Parameters	Cochez cette case pour spécifier la valeur des paramètres **Load Operator**, **Data Connector**, **Job Name**, et **Layout Name (schema)** qui seront utilisés lors de la génération d'un script durant l'exécution du Job. Si vous ne les spécifiez pas manuellement, le système utilisera leurs valeurs par défaut.
	Load Operator	Spécifiez l'opérateur de chargement. Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée.
	Data Connector	Spécifiez le connecteur de données. Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée.
	Job Name	Spécifiez le nom du Job TPT (Teradata Parallel Transporter) défini via la commande Teradata *tbuild*. Pour plus d'informations concernant la commande *tbuild*, consultez http://developer.teradata.com/sites/all/files/documentation/linked_docs/html/online/B035-2436-088A/wwhelp/wwhimpl/js/html/wwhelp.htm (en anglais). Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée.
	Layout Name(schema)	Spécifiez un schéma pour le chargement des données. Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée.
	Define Log table	Cochez cette case pour spécifiez le nom de table de log.
	Return mload error	Cochez cette case pour spécifier le code de retour à partir duquel retourner une erreur dans la console.
	Define character set	Cochez cette case pour spécifier l'encodage des caractères à utiliser dans votre système.
	Apply TPT consumer operator optional attributes	Cochez cette case afin de définir les attributs facultatifs pour l'opérateur du consommateur sélectionné. Pour plus d'informations concernant les attributs facultatifs de chaque opérateur de consommateur, consultez le Guide *Teradata Parallel Transporter Reference* (en anglais) à l'adresse suivante : http://developer.teradata.com/sites/all/files/documentation/linked_docs/2436020A_TPT-Reference-13.10.pdf.
	Optional attributes	Cliquez sur le bouton **[+]** sous la table pour ajouter autant des lignes que nécessaire pour chaque attribut facultatif. Configurez les paramètres comme suit pour chaque attribut :

		• **Name** : cliquez dans la cellule et sélectionnez un attribut facultatif dans la liste déroulante. La liste des attributs facultatifs varie selon l'opérateur du consommateur sélectionné. Pour plus d'informations concernant les attributs facultatifs pour chaque opérateur du consommateur, consultez *Attributs facultatifs supportés pour chaque opérateur de consommateur*. • **Value** : saisissez la valeur de l'attribut facultatif. Pour plus d'informations concernant la définition de la valeur d'un attribut, consultez le Guide *Teradata Parallel Transporter Reference* (en anglais) à l'adresse suivante : http://developer.teradata.com/sites/all/files/documentation/linked_docs/2436020A_TPT-Reference-13.10.pdf. Cette table apparaît uniquement lorsque la case **Apply TPT consumer operator optional attributes** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**EXIT_VALUE** : code de sortie du processus. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Précédé par un composant d'entrée, le **tTeradataTPTUtility** écrit les données entrantes dans un fichier puis les charge du fichier dans la base de données Teradata. Ce traitement se fait avec de hautes performances. Pour plus d'informations concernant l'utilisation de ce composant, consultez http://developer.teradata.com/sites/all/files/documentation/linked_docs/html/online/B035-2436-088A/wwhelp/wwhimpl/js/html/wwhelp.htm (en anglais).	
Limitation	Il est nécessaire d'installer l'outil de client Teradata avec les fonctionnalités Teradata Parallel Transporter Base et Teradata Parallel Transporter Stream sur la machine où les Jobs utilisant ce composant sont exécutés.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Charger des données dans une base de données Teradata*.

tTeradataTPump

Propriétés du tTeradataTPump

Famille de composant	Databases/Teradata	
Fonction	Ce composant insère, met à jour ou supprime des données dans la base de données Teradata à l'aide de l'utilitaire de chargement TPump qui permet d'obtenir des données en temps quasi réel dans un entrepôt de données.	
Objectif	Utilisez ce composant pour insérer, mettre à jour ou supprimer des données dans la base de données Teradata, en particulier dans les environnements où les fenêtres de lot sont réduites et où la maintenance des entrepôts coïncident avec les heures normales de travail.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Execution platform	Sélectionnez le type de système d'exploitation que vous utilisez.
	Host	Nom de l'hôte ou l'adresse IP du serveur de la base de données.
	Database name	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant,

		sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Script generated folder	Parcourez votre répertoire et sélectionnez la destination du fichier qui sera créé.
	Action to data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations **Update** et **Delete**. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (**Advanced settings**) pour une utilisation optimisée de ces opérations.*
	Where condition in case Delete	Saisissez une condition, qui, si elle est vérifiée, supprimera la ligne. Ce champ apparaît uniquement lorsque **Delete** est sélectionné dans la liste déroulante **Action to data**.
	Load file	Entrez le nom du fichier à partir duquel vous souhaitez charger les données.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Error file	Parcourez votre répertoire et sélectionnez la destination du fichier dans lesquels seront enregistrés les messages d'erreur.
Advanced settings	*Define Log table*	Cochez cette case pour définir la table de log que vous souhaitez utiliser au lieu de celle par défaut, à savoir celle que vous avez définie dans l'onglet **Basic settings**. La syntaxe requise pour définir la table de log est *databasename.logtablename*.
	BEGIN LOAD	Ce champ vous permet de définir votre commande BEGIN LOAD pour lancer ou relancer une tâche TPump. Vous pouvez spécifiez le nombre de sessions à utiliser, la limite d'erreurs, ainsi que tout autre paramètre nécessaire à l'exécution de la tâche. La valeur par défaut est : `SESSIONS 8 PACK 600 ARRAYSUPPORT ON CHECKPOINT 60 TENACITY 2 ERRLIMIT 1000.` Pour plus d'informations, consultez la documentation *Teradata Parallel Data Pump Reference*.
	Return tpump error	Cochez cette case pour spécifier le code de retour à partir duquel retourner une erreur dans la console.
	Define character set	Spécifiez l'encodage des caractères dont vous avez besoin pour utiliser votre système.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**EXIT_VALUE** : code de sortie du processus. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Teradata.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Insérer des données dans une table d'une base de données Teradata

Dans ce scénario, l'objectif est de créer un Job utilisant le composant **tTeradataTPump** pour insérer des données clients dans une table d'une base de données Teradata et retourner la valeur de retour lorsqu'une erreur se produit.

Trois composants sont utilisés pour implémenter cette tâche :

- **tRowGenerator** : génère autant de lignes que nécessaire, en utilisant au hasard des données clients prises dans une liste.
- **tFileOutputDelimited** : écrit les données clients dans un fichier délimité.
- **tTeradataTPump** : insère les données clients dans la table de la base de données Teradata en mode Tpump.

Déposer les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tRowGenerator**, **tFileOutputDelimited**, **tTeradataTPump**.
2. Liez le **tRowGenerator** au **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main**.
3. Liez le **tRowGenerator** au **tTeradataTPump** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Configurer les composants

1. Double-cliquez sur le **tRowGenerator** pour ouvrir la fenêtre **[RowGenerator Editor]**.

   Dans la fenêtre **[RowGenerator Editor]**, définissez les données à générer. Pour ce Job, le schéma se compose de deux colonnes : *ID* et *Name*.

   Dans le champ **Number of Rows for RowGenerator**, saisissez le nombre de lignes à générer.

2. Double-cliquez sur le **tFileOutputDelimited** afin d'afficher sa vue **Component** et définir ses propriétés.

3. A côté du champ **File Name**, parcourez votre répertoire jusqu'au fichier de sortie, ou saisissez un nom pour le fichier à créer.

4. Entre guillemets doubles, saisissez les séparateurs à utiliser, à côté des champs **Row Separator** et **Field Separator**, par exemple *\n* et un point-virgule.

Cliquez sur **Edit schema** et vérifiez que le schéma est retrouvé à partir du schéma d'entrée. Au besoin, cliquez sur **Sync Columns**.

5. Double-cliquez sur le composant **tTeradataTPump** afin d'ouvrir sa vue **Component**.

   Dans l'onglet **Basic settings** de la vue **Component**, définissez les paramètres du composant **tTeradataTPump**.

6. Saisissez le nom de la base de données dans le champ **Database name**, votre identifiant de connexion à la base de données dans le champ **User name**, et votre mot de passe dans le champ **Password**.

7. Spécifiez la table dans laquelle insérer les données clients. Dans ce scénario, la table est *mytable*.

8. Dans les champs **Script generated folder**, **Load file** et **Error file**, spécifiez le répertoire respectivement au dossier dans lequel vous stockez les fichiers de scripts générés.

9. Dans le champ **Load File**, spécifiez le fichier contenant les données clients à insérer.

10. Dans le champ **Error file**, spécifiez le fichier contenant les informations concernant les erreurs.

11. Dans le champ **Action on data**, sélectionnez l'option **Insert**.

Exécuter le Job

1. Appuyez sur **F6** pour exécuter le Job.

2. La console de la vue **Run** s'affiche ainsi :

```
Running job 'tpump'...
Starting job tpump at 17:01 19/07/2010.

[statistics] connecting to socket on port 3740
[statistics] connected
[statistics] disconnected
Job tpump ended at 17:02 19/07/2010. [exit code=0]
```

3. Double-cliquez sur le composant **tTeradataTPump** pour retourner à sa vue **Component**.

4. Dans l'onglet **Advanced settings**, cochez la case **Return tpump error** et saisissez le code de retour à partir duquel retourner une erreur dans la console. Dans cet exemple, saisissez le chiffre *4* et utilisez les valeurs par défaut pour les autres paramètres.

5. Appuyez sur **F6** pour exécuter le Job.

6. La console de la vue **Run** s'affiche ainsi :

```
Running job 'tpump'...
Starting job tpump at 17:56 19/07/2010.

[statistics] connecting to socket on port 3549
[statistics] connected
[statistics] disconnected
Exception in component tTeradataTPump_1
java.lang.RuntimeException: TPump returned exit code 12
    at
bug.tpump_0_1.tpump.tTeradataTPump_1Process(tpump.java:307)
    at bug.tpump_0_1.tpump.runJobInTOS(tpump.java:504)
    at bug.tpump_0_1.tpump.main(tpump.java:378)
Job tpump ended at 17:56 19/07/2010. [exit code=0]
```

Une erreur se produit et le `TPump` retourne le code de retour, de 12. Si vous avez besoin d'informations détaillées concernant les informations détaillées de l'erreur, vous pouvez ouvrir le fichier de log stocké dans le répertoire spécifié dans le champ **Error file** et dans l'onglet **Basic settings** de la vue **Component**.

tVectorWiseCommit

Propriétés du tVectorWiseCommit

Famille de composant	Databases/VectorWise	
Fonction	Le composant **tVectorWiseCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tVectorWiseConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tVectorWiseCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants VectorWise et notamment avec les composants **tVectorWiseConnection** et **tVectorWiseRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tVectorWiseCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tVectorWiseConnection

Propriétés du tVectorWiseConnection

Famille de composant	Databases/VectorWise	
Fonction	Le composant **tVectorWiseConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Server	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement

		après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec des composants VectorWise, notamment les composants **tVectorWiseCommit** et **tVectorWiseRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au composant **tVectorWiseConnection**, consultez *tMysqlConnection*.

tVectorWiseInput

Propriétés du tVectorWiseInput

Famille de composant	Databases/ VectorWise	
Fonction	Le composant **tVectorWiseInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tVectorWiseInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Server	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.

		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table name	Saisissez le nom de la table.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Guess schema	Cliquez sur le bouton pour récupérer le schéma de la table.
Advanced settings	*Trim all the String/Char columns*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Dans la colonne **Column**, saisissez le nom de la colonne dans laquelle supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données VectorWise.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez les scénarios du composant générique **tDBInput** :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*
- *Scénario 2 : Utiliser la variable StoreSQLQuery*

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du **tContextLoad**.

tVectorWiseOutput

Propriétés du tVectorWiseOutput

Famille de composant	Databases/ VectorWise	
Fonction	Le composant **tVectorWiseOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	**Le tVectorWiseOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire.

		Clear a table : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.

		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tTeradataOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Support null in "SQL WHERE" statement	Cochez cette case pour prendre en compte les valeurs Null d'une table de base de données. Assurez-vous que la case **Nullable** est bien cochée pour les colonnes du schéma correspondantes.
	Use Batch Size	Cochez cette case pour activer le mode batch (par lots) afin d'améliorer les performances du traitement de données lorsque vous avez besoin d'insérer un grand nombre d'enregistrements. Cette option n'est supportée que sur la version 2.5 et plus de Vectorwise, et n'est disponible que si vous sélectionnez **Insert** dans la liste déroulante **Action on data** dans l'onglet **Basic settings**.
	Batch Size	Spécifiez le nombre d'enregistrements à envoyer à la base de données par lot. Ce champ apparaît uniquement lorsque la case **Use Batch Size** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données VectorWise. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tVectorWiseRollback

Propriétés du tVectorWiseRollback

Famille de composant	Databases/VectorWise	
Fonction	Le **tVectorWiseRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tVectorWiseConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants VectorWise, notamment les composants **tVectorWiseConnection** et **tVectorWiseCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tVectorWiseRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille*.

tVectorWiseRow

Propriétés du tVectorWiseRow

Famille de composant	Databases/ VectorWise	
Fonction	Le **tVectorWiseRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tVectorWiseRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à traiter.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Query	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.	

Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tVerticaBulkExec

Propriétés du tVerticaBulkExec

Les composants **tVerticaOutputBulk** et **tVerticaBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tVerticaOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Famille de composant	Databases/ Vertica	
Fonction	Le composant **tVerticaBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tVerticaBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Vertica.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez dans la liste la version de Vertica que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma de la base de données à utiliser.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore.

		Create table if not exists : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	File Name	Nom du fichier à charger. Ce fichier doit se situer sur la même machine que le *Studio Talend* ou que le Job contenant un **tVerticaBulkExec**, afin que le composant fonctionne correctement.
Advanced settings	*Additional JDBC Parameters*	Ajoutez des paramètres JDBC supplémentaires nécessaires à la connexion à la base de données. Cette option n'est pas disponible lorsque vous cochez la case **Use an existing connection** dans la vue **Basic settings**.
	Action on data	Sur les données de la table définie, vous pouvez effectuer les opérations suivantes : • **Bulk insert** : Ajouter plusieurs entrées à la table. Si des doublons sont trouvés, le Job s'arrête. • **Bulk update** : Effectuer des modifications simultanées sur différentes entrées.
	Write to ROS (Read Optimized Store)	Cochez cette case pour stocker les données dans une mémoire physique, et ainsi optimiser la lecture de ces données, puisqu'elles seront compressées et pré-triées.
	Exit job if no row was loaded	Le Job s'arrête automatiquement si aucune ligne n'a été chargée.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs.
	Null string	Chaîne de caractères affichée pour signifier que la valeur est nulle.
	Rejected data file	Spécifiez le fichier dans lequel vous souhaitez que soient écrites les lignes rejetées.
	Exception log file	Spécifiez le fichier dans lequel vous souhaitez que soit écrite une explication pour chaque ligne rejetée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.

Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**ACCEPTED_ROW_NUMBER** : le nombre de lignes chargées dans la base de données. **REJECTED_ROW_NUMBER** : le nombre de lignes rejetées. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé avec un composant **tVerticaOutputBulk**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données Vertica.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénarios associés

Pour un scénario associé au composant **tVerticaBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du composant **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du composant **tOracleBulkExec**.

tVerticaClose

Propriétés du tVerticaClose

Famille de composant	Databases/Vertica	
Fonction	Le composant **tVerticaClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Vertica.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tVerticaConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénario associé

Ce composant est étroitement lié aux composants **tVerticaConnection** et **tVerticaRollback**. Il est généralement utilisé avec un composant **tVerticaConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tVerticaClose**, consultez *tMysqlConnection*.

tVerticaCommit

Propriétés du tVerticaCommit

Famille de composant	Databases/Vertica	
Fonction	Le composant **tVerticaCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tVerticaConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tVerticaCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Vertica et notamment avec les composants **tVerticaConnection** et **tVerticaRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tVerticaCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tVerticaConnection

Propriétés du tVerticaConnection

Famille de composant	Databases/Vertica	
Fonction	Le composant **tVerticaConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez dans la liste la version de Vertica que vous utilisez.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Additional JDBC Parameters	Ajoutez des paramètres JDBC supplémentaires nécessaires à la connexion à la base de données.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications

		doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec des composants Vertica, notamment les composants **tVerticaCommit** et **tVerticaRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénario associé

Pour un scénario associé au composant **tVerticaConnection**, consultez *tMysqlConnection*.

tVerticaInput

Propriétés du tVerticaInput

Famille de composant	Databases/ Vertica	
Fonction	Le composant **tVerticaInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tVerticaInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (**Main row**).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez dans la liste la version de Vertica que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à lire.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Additional JDBC Parameters*	Ajoutez des paramètres JDBC supplémentaires nécessaires à la connexion à la base de données. Cette option n'est pas disponible lorsque vous cochez la case **Use an existing connection** dans la vue **Basic settings**.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Vertica.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de	

	la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénarios associés

Voir également les scénarios du **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Voir également le scénario du **tContextLoad,**, consultez *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement*.

tVerticaOutput

Propriétés du tVerticaOutput

Famille de composant	Databases/ Vertica	
Fonction	Le composant **tVerticaOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tVerticaOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez dans la liste la version de Vertica que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	La liste **Action on table** est indisponible si vous cochez la case **Enable parallel execution** dans la vue **Advanced settings**. Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **Default** : n'effectuer aucune opération de table.

		Drop and create table : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table.
	Use "drop cascade"	Cette case est disponible lorsque vous sélectionnez l'option relative à la suppression de table dans la liste **Action on table**. Cochez cette case pour supprimer tous les objets liés à la table qui va être supprimée.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. **Copy** : Lire des données d'un fichier texte et insérer des tuples dans le système de stockage optimisé en écriture (Write Optimized Store) ou directement dans le système de stockage optimisé en lecture (Read Optimized Store). Cette option est idéale pour le chargement de masse. Pour plus d'informations, consultez votre manuel Vertica SQL. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si

		vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC Parameters*	Ajoutez des paramètres JDBC supplémentaires nécessaires à la connexion à la base de données. Cette option n'est pas disponible lorsque vous cochez la case **Use an existing connection** dans la vue **Basic settings**.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
Paramètres Copy Cette zone est uniquement disponible si vous avez choisi **Copy** dans la liste **Action on data**. Pour des informations plus détaillées concernant les paramètres **Copy**, consultez votre Manuel de Référence Vertica SQL.	*Abort on error*	Cochez cette case afin d'arrêter l'opération de copie (**Copy)** sur les données si une ligne est rejetée et annule cette opération. Ainsi, aucune donnée n'est traitée.
	Maximum rejects	Saisissez un nombre afin de configurer la commande REJECTMAX utilisée par Vertica, qui indique la limite maximale du nombre d'enregistrements logiques à rejeter avant qu'un chargement n'échoue. Si ce nombre n'est pas spécifié ou si sa valeur est 0, le nombre de rejets autorisés est illimité.
	No commit	Cochez cette case pour empêcher l'opération actuelle de commiter automatiquement.
	Exception file	Saisissez le chemin d'accès, ou parcourez et sélectionnez le fichier dans lequel sont écrits les messages indiquant le nombre de lignes d'entrée et la raison de chaque rejet d'enregistrement de données.
	Exception file node	Indiquez quel est le nœud du fichier d'exception. Si ce champ n'est pas renseigné, le nœud de l'initiateur de la requête est choisi par défaut.
	Rejected data file	Saisissez le chemin d'accès, ou parcourez et sélectionnez le fichier dans lequel seront écrites les lignes rejetées. Ce fichier pourra ensuite être modifié afin de résoudre des problèmes, et pourra être regénéré.
	Rejected data file node	Indiquez quel est le nœud du fichier des données rejetées. Si ce champ n'est pas renseigné, le nœud de l'initiateur de la requête est choisi par défaut.
	Use Batch mode	Cochez cette case afin d'activer le mode batch pour le traitement des données. Dans le champ **Batch Size** qui apparaît, vous pouvez saisir le nombre de lignes par lot à charger. Cette case est disponible uniquement si vous avez choisi l'option **Insert**, **Update**, **Single Insert Query** ou **Delete** dans le champ **Action on data**.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.

		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tVerticaOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	Support null in "SQL WHERE" statement	Cochez cette case afin d'autoriser les valeurs nulles dans l'instruction "SQL WHERE".
	Create projection when create table	Cochez cette case afin de créer une projection pour la table à créer. Cette case est disponible uniquement lorsque vous avez sélectionné l'option de création de table dans la liste **Action on table** de la vue **Basic settings**. Elle disparaît lorsque vous cochez la case **Enable parallel execution**, dans la vue **Advanced settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_COPIED** : nombre de lignes copiées. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Vertica. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénarios associés

Pour un scénario associé au composant **tVerticaOutput**, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tVerticaOutputBulk

Propriétés du tVerticaOutputBulk

Les composants **tVerticaOutputBulk** et **tVerticaBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tVerticaOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/Vertica	
Fonction	Ecrit un fichier composé de colonnes et basé sur le délimiteur défini et sur les standards Vertica.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Vertica.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer. *Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.*
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les

		modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Row separator*	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec le composant **tVerticaBulkExec**. Ensemble, ils offrent un gain de performance important pour l'alimentation d'une base de données Vertica.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénarios associés

Pour un scénario associé au **tVerticaOutputBulk**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tVerticaOutputBulkExec

Propriétés du tVerticaOutputBulkExec

Les composants **tVerticaOutputBulk** et **tVerticaBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tVerticaOutputBulkExec**.

Famille de composant	Databases/Vertica	
Fonction	Le composant **tVerticaOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tVerticaOutputBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données Vertica.	
Basic settings	*Property Type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Sélectionnez dans la liste la version de Vertica que vous utilisez.
	Use existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	DB Name	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle. **Create table** : créer une table qui n'existe pas encore. **Create table if not exists** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table.

	Schema and *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
	File Name	Nom du fichier à générer et à charger. Ce fichier est généré sur la même machine que le *Studio Talend* ou que le Job contenant le **tVerticaOutputBulkExec** puis est chargé dans la base de données spécifiée dans le champ **Host**.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
Advanced settings	*Additional JDBC Parameters*	Ajoutez des paramètres JDBC supplémentaires nécessaires à la connexion à la base de données. Cette option n'est pas disponible lorsque vous cochez la case **Use an existing connection** dans la vue **Basic settings**.
	Write to ROS (Read Optimized Store)	Cochez cette case pour stocker les données dans une mémoire physique, et ainsi optimiser la lecture de ces données, puisqu'elles seront compressées et pré-triées.
	Exit job if no row was loaded	Le Job s'arrête automatiquement si aucune ligne n'a été chargée.
	Field Separator	Caractère, chaîne ou expression régulière séparant les champs.
	Null string	Chaîne de caractères affichée pour signifier que la valeur est nulle.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Rejected data file	Spécifiez le fichier dans lequel vous souhaitez que soient écrites les lignes rejetées.
	Exception log file	Spécifiez le fichier dans lequel vous souhaitez que soit écrite une explication pour chaque ligne rejetée.

	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ACCEPTED_ROW_NUMBER** : le nombre de lignes chargées dans la base de données. **REJECTED_ROW_NUMBER** : le nombre de lignes rejetées. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénarios associés

Pour un scénario associé au **tVerticaOutputBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du composant **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.

tVerticaRollback

Propriétés du tVerticaRollback

Famille de composant	Databases/Vertica	
Fonction	Le **tVerticaRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tVerticaConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants Vertica, notamment les composants **tVerticaConnection** et **tVerticaCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénario associé

Pour un scénario associé au composant **tVerticaRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille*.

tVerticaRow

Propriétés du tVerticaRow

Famille de composant	Databases/ Vertica	
Fonction	Le **tVerticaRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tVerticaRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB version	Sélectionnez la version de Vertica que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table name	Nom de la table à lire.
	Query type	Peut être Built-in ou Repository.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.
	Query	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC Parameters*	Ajoutez des paramètres JDBC supplémentaires nécessaires à la connexion à la base de données. Cette option n'est pas disponible lorsque vous cochez la case **Use an existing connection** dans la vue **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Conseils et techniques

Utilisés ensemble, le *Studio Talend* et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Scénario associé

Pour un scénario associé au composant **tVerticaRow**, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du **tMysqlRow**.

Databases - Autres composants

Le chapitre *Databases - Autres composants* décrit les connecteurs donnant accès à un éventail de bases de données et fournissant les outils de gestion de bases de données. Ces connecteurs couvrent divers besoins : ouverture de connexion, lecture et écriture de tables, commit de transactions, ainsi que des rollbacks pour le traitement des erreurs. Ces composants sont présents dans la famille **Databases** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

D'autres types de connecteurs de bases de données, tels que les connecteurs de bases de données traditionnelles et de bases de données d'appliance, de datawarehouse, de bases de données verticales, sont documentés dans le *Databases - Composants traditionnels* et le *Databases - Composants d'appliance et de datawarehouse*.

tCreateTable

Propriétés du tCreateTable

Ce composant se trouve à la racine de la famille **Databases** de la **Palette** du *Studio Talend*. Le composant **tCreateTable** couvre des besoins indirectement liés à l'utilisation des bases de données de tout type.

Famille de composant	Databases	
Fonction	Le **tCreateTable** crée, supprime ou supprime et recrée une table spécifique.	
Objectif	Ce composant permet de créer ou de supprimer n'importe quelle table d'une base de données	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Database Type	Sélectionnez le type de SGBD dans la liste proposée. Les propriétés peuvent différer légèrement en fonction du type de base de données sélectionné. Les champs additionnels sont définis dans ce tableau, à la suite des propriétés générales.
	DB Version	Sélectionnez la version de la base de données que vous utilisez.
	Table Action	Sélectionnez l'action à effectuer sur la base de données : **Create table** : lorsque vous savez que la table n'existe pas déjà. **Create table when not exists** : lorsque vous ne savez pas si la table existe déjà. **Drop and create table** : lorsque vous savez que la table existe déjà et que vous souhaitez la remplacer.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database name	Nom de la base de données.
	Schema	Nom du schéma. (DB2, Greenplum, Informix, Oracle, PostgresPlus, Postgresql, Vertica)
	Username and *Password*	Informations d'authentification de l'utilisateur de base de données.

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table name	Saisissez entre guillemets un nom pour la table nouvellement créée.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant uniquement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log lors du traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
	Additional JDBC Parameters	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Cette option n'est pas disponible si vous avez coché la case **Use an existing connection** dans la vue **Basic settings** (AS/400, MSSQL Server).
	Create projection	Cochez cette case pour créer une projection (Vertica).
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes de base de données et couvre toutes les possibilités des requêtes SQL. Plus de scénarios sont disponibles pour chaque composant d'entrée de type base de données.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Champs spécifiques à certaines bases de données :

Access	*Access File*	Nom et chemin d'accès du fichier à traiter.
Firebird	*Firebird File*	Nom et chemin d'accès du fichier à traiter.
HSQLDb	*Running Mode*	Sélectionnez dans la liste le type de serveur correspondant aux paramètres de votre base de données..
	Use TLS/SSL Sockets	Cochez cette case pour autoriser, si besoin, le mode sécurisé.
	DB Alias	Nom de la base de données.
	Case Sensitive	Cochez cette case pour rendre sensible à la casse le nom de la table/colonne.
Interbase	*Interbase File*	Nom et chemin d'accès du fichier à traiter.
JavaDb	*Framework Type*	Sélectionnez dans la liste un framework pour votre base de données.
	Structure type	Sélectionnez dans la liste le type de structure.
	DB Root Path	Parcourez votre système jusqu'à la racine de votre base de données.
Mysql	*Temporary table*	Cochez cette case si vous souhaitez sauvegarder temporairement la table créée.
ODBC	*ODBC Name*	Nom de la base de données.
Oracle	*Connection Type*	Liste déroulante des pilotes disponibles: **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. **Service Oracle** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante. **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle. **WALLET** : Sélectionnez ce type de connexion pour stocker les informations d'identification dans un portefeuille Oracle (Oracle wallet).
SQLite	*SQLite File*	Nom et chemin d'accès au fichier à traiter. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.
Teradata	*Create*	Sélectionnez la table à partir de la liste déroulante. La table à créer peut être : -**SET TABLE** : table ne permettant pas de dupliquer les lignes -**MULTI SET TABLE** : table permettant de dupliquer les lignes.

Scénario : Créer une nouvelle table dans une base de données MySQL

Le Job décrit ci-dessous a été conçu pour créer une nouvelle table dans une base de données, composée d'un schéma pris à partir du schéma d'un fichier délimité stocké dans le **Repository**. Ce Job est formé d'un seul composant.

1. Cliquez et déposez un composant **tCreateTable** de la famille **Databases** de la **Palette** dans l'espace de modélisation graphique.

2. Dans la liste **Database type** de l'onglet **Basic settings** de la vue **Component**, sélectionnez *Mysql*.

3. Dans la liste déroulante **Table Action**, sélectionnez **Create table**.

4. Ne cochez la case **Use Existing Connection** uniquement si vous utilisez un composant avec une connexion dédiée (pour plus d'informations, consultez *tMysqlConnection*). Vous n'utiliserez pas cette option pour cet exemple.

5. Dans la liste déroulante **Property type**, sélectionnez **Repository** pour que tous les champs de connexion à la base de données soient renseignés automatiquement. Si vous n'avez pas configuré de connexion aux bases de données dans le dossier **Metadata** du répertoire **DB connection**, renseignez manuellement ces informations de connexion après avoir sélectionné **Built-in**.

6. Dans le champ **Table Name**, saisissez le nom de la nouvelle table à créer.

7. Si vous souhaitez récupérer le schéma d'une métadonnée (sans que celle-ci ne soit une connexion à une base de données), sélectionnez **Repository** puis la métadonnée souhaitée.

8. Dans tous les cas (**Built-in** ou **Repository**), cliquez sur **Edit Schema** pour vérifier le type de données dans la colonne **DB Type**. Cliquez sur **Edit schema** pour définir la structure des données.

9. Cliquez sur le bouton **Reset DB Types** si la colonne **DB type** est vide ou affiche des marques de différence (couleur orange). Cela permet de donner à n'importe quel type de données le type de données adéquat. Cliquez sur **OK** pour valider vos modifications et fermer la boîte de dialogue.

10. Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

La table créée est vide mais elle contient toutes les colonnes définies dans le schéma.

tDBInput

Propriétés du tDBInput

Famille de composant	Databases/DB Generic	
Fonction	**Le composant tDBInput** lit une base de données et en extrait des champs à l'aide de requêtes. Pour utiliser ce composant, les pilotes ODBC du SGBD doivent être installés et les connexions ODBC doivent être configurées via l'assistant de configuration de connexion à une base de données.	
Objectif	Le **tDBInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row). Pour des raisons de performances, préférez toujours un composant Input spécifique au composant générique.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Database	Nom de la source de données définie via l'assistant de configuration de connexion à une base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table source dans laquelle capturer les modifications apportées aux données.
	Query type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
Advanced settings	*Additional JDBC parameters*	Spécifiez les propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL grâce à une connexion générique ODBC.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Afficher les données sélectionnées à partir d'une table

Le Job suivant, à deux composants, lit les données d'une base via une requête et affiche dans la console (vue **Run**) les données délimitées.

Avant de reproduire ce Job, le pilote ODBC de MySQL doit avoir été installé et la connexion ODBC correspondante doit avoir été configurée.

1. Cliquez et déposez les composants **tDBInput** et **tLogRow** dans l'espace de modélisation graphique.

2. Cliquez-droit sur le **tDBInput** et, en gardant le bouton droit enfoncé, glissez jusqu'au **tLogRow** pour créer une connexion de type **Main Row**.

3. Double-cliquez sur le **tDBInput** pour ouvrir sa vue **Component** et renseignez les propriétés du composant :

4.

5. Renseignez les informations de connexion à la base de données dans les champs **Database**, **Username** et **Password**.

6. Cliquez sur le bouton **Edit Schema** et répertoriez les codes des magasins dans une colonne et les ventes dans une autre.

7. Renseignez le nom de la table dans le champ **Table Name**.

8. Saisissez la requête à exécuter en veillant à ce que l'ordre des colonnes correspondent à celui fourni dans le schéma. Dans cet exemple, comme toutes les colonnes du schéma ont été sélectionnées, il est possible d'utiliser l'astérisque.

9. De la même façon, cliquez sur le deuxième composant pour le définir.

10. Indiquez le séparateur de champs : il s'agit ici d'une barre verticale.

11. Passez sur la vue **Run** et cliquez sur le bouton **Run** pour exécuter le Job.

La base de données est alors analysée. Les données recherchées sont extraites et sont affichées directement sur la console (vue **Run**), où vous pouvez les voir.

Scénario 2 : Utiliser la variable StoreSQLQuery

La variable StoreSQLQuery peut être utilisée pour déboguer un scénario **tDBInput** qui ne s'exécuterait pas correctement. Elle permet d'alimenter la requête de votre composant **tDBInput** de manière dynamique.

1. Partez du scénario 1 décrit ci-dessus, auquel vous ajouterez un composant **tJava**.

2. Reliez les composants **tDBInput** et **tJava** via une connexion Trigger de type **OnComponentOk**. Dans le cas présent, faites en sorte que la commande du **tDBInput** s'exécute avant la commande du **tJava**.

3. Configurez les composants **tDBInput** et **tLogRow** comme dans le scénario 1 décrit plus haut.

4. Passez à la vue appelée **Context**.

5. Créez un nouveau paramètre que vous appellerez **StoreSQLQuery** et donnez-lui une valeur de 1 par défaut. Cette valeur indique que StoreSQLQuery est valide (**true**) et pourra être utilisé par la variable globale QUERY.

6. Cliquez sur le composant **tJava** et affichez la vue **Component**. Tapez la commande `System.Out.println("")` pour afficher le contenu de la requête, puis, dans les parenthèses, appuyez sur **Ctrl+Espace** pour accéder à la liste des variables, et sélectionnez la variable globale QUERY.

7. Affichez la vue **Run** et exécutez le Job.

8. La requête associée au composant **tDBInput** s'affiche dans la console (vue **Run**), à la fin des résultats du Job :

```
971|9143650232
978|1475829234
989|3139722244
SELECT * FROM   comprehensive
Job tDBInputodbc ended at 10:15 04/03/2010. [exit code=0]
```

tDBOutput

Propriétés du tDBOutput

Famille de composant	Databases	
Fonction	Le composant **tDBOutput** écrit, met à jour, modifie ou supprime les données d'une base de données. Pour utiliser ce composant, les pilotes ODBC du SGBD doivent être installés et les connexions ODBC doivent être configurées via l'assistant de configuration de connexion à une base de données.	
Objectif	Le **tDBOutput** exécute l'action définie sur les données d'une table, en fonction du flux entrant provenant du composant précédent. Préférez toujours un composant Output spécifique au composant générique.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré.

		Delete : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Clear data in table	Cochez cette case pour que les données soient supprimées de la table sélectionnée avant toute action.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Additional JDBC parameters*	Spécifiez les propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de transaction

		(cependant pas de rollback) et surtout une meilleure performance des exécutions.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tDBOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Use java.sql.Statement	Cochez cette case pour utiliser l'objet `Statement`, si l'objet `PreparedStatement` n'est pas supporté par votre SGBD.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il doit être utilisé en tant que composant de sortie. Il permet d'effectuer des actions sur les données d'une table d'une base de données. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC

Ce scénario efface les données dans une table d'une base de données MySQL puis ajoute une ligne à cette table.

La table, nommée *Date*, contient une colonne nommée *date*, de type date.

Avant de reproduire ce Job, le pilote ODBC de MySQL doit avoir été installé et la connexion ODBC correspondante doit avoir été configurée.

1. Déposez un **tDBOutput** et un **tRowGenerator** de la **Palette** dans l'espace de modélisation graphique.
2. Connectez les composants à l'aide d'un lien **Row** > **Main**.

3. Double-cliquez sur le **tRowGenerator** pour ouvrir son **Schema editor**.

4. Cliquez sur le bouton **[+]** pour ajouter une ligne.

   Saisissez *date* comme nom de colonne.

   Sélectionnez *Date* dans la liste des types de données.

   Sélectionnez *getCurrentDate* dans la liste **Functions**.

   Saisissez *1* dans le champ **Number of Rows for RowGenerator**, une seule ligne sera ajoutée à la table.

   Cliquez sur **OK** pour fermer l'éditeur et propager les modifications au **tDBOutput**.
5. Double-cliquez sur le **tDBOutput** pour ouvrir sa vue **Basic settings**.

6. Dans le champ **Database**, saisissez le nom de la source de données définie durant la configuration de la connexion ODBC MySQL.

   Pour configurer une connexion ODBC, cliquez sur le bouton pour ouvrir l'assistant de configuration de la connexion à la base de données.

7. Dans les champs **Username** et **Password**, saisissez les informations d'authentification à la base de données.

8. Dans le champ **Table**, saisissez le nom de la table, *Date* dans cet exemple.

9. Dans le champ **Action on data**, sélectionnez *Insert* afin d'insérer une ligne dans la table.

10. Cochez la case **Clear data in table** afin de vider la table avant l'insertion.

11. Sauvegardez le Job et appuyez sur **F6** pour l'exécuter.

date
2012-04-12 14:49:54
(NULL)

   Comme affiché ci-dessus, la table ne contient plus qu'une ligne concernant la date et l'heure courantes.

tDBSQLRow

Propriétés du tDBSQLRow

Famille de composant	Databases / DB Generic	
Fonction	**tDBSQLRow** est le composant générique à tout type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie. Pour une question de performances, préférez toujours un composant de base de données spécifique au composant générique.	
Objectif	Selon la nature de la requête et de la base de données, **tDBSQLRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes. Pour utiliser ce composant, les pilotes ODBC du SGBD doivent être installés et les connexions ODBC doivent être configurées via l'assistant de configuration de connexion à une base de données.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Datasource	Nom de la source de données définie dans l'assistant de configuration de la connexion à la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table de base de données à modifier.
	Query type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur. Si nécessaire, vous pouvez récupérer les lignes en erreur via un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez les propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Vous pouvez configurer les paramètres d'encodage dans ce champ.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL. Utilisez le composant DBRow correspondant au type de base de données dont vous vous servez. La plupart des bases de données ont leur propre composant DBRow spécifique.
Prérequis	ODBC (Open DataBase Connectivity) est un protocole standard permettant d'accéder aux informations de serveurs de bases de données. Vous devez préalablement créer une source de données (datasource) ODBC et la relier à votre base de données pour en lire les informations. Pour créer une source ODBC sous Windows, par exemple, allez à **Panneau de configuration** > **Outils d'administration** > **Sources de données (OBCD)** puis renseignez les propriétés de la source à partir de l'assistant.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Réinitialiser l'auto-incrémentation d'une base de données

Le Job suivant, à un seul composant, permet de réinitialiser l'auto-incrémentation de la base de données (valeur initiale = 1). Il ne permet pas de générer un flux en sortie car il est généralement utilisé avant l'exécution d'un script.

Avant de reproduire ce Job, le pilote ODBC du SGBD correspondant doit avoir été installé et la connexion ODBC correspondante doit avoir été configurée.

1. Glissez-déposez le composant **tDBSQLRow** de la **Palette** dans l'espace graphique de modélisation.

2. Double-cliquez sur le **tBDSQLRow** afin d'ouvrir sa vue **Basic settings**.

3. Dans le champ **Property Type**, sélectionnez **Repository** puisque la connexion ODBC a été configurée et sauvegardée dans le **Repository**. Les champs suivants sont automatiquement renseignés.

   Pour plus d'informations concernant le stockage des connexions aux bases de données dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

4. Le champ **Schema** est sur **Built-in** pour ce Job et n'est pas vraiment important dans cet exemple particulier puisque l'action se fait sur l'auto-incrémentation de la table et non sur les données.

5. Dans la liste **Query type,** sélectionnez l'option **Built-in**. Cliquez sur le bouton **[...]** à côté du champ **Query** pour ouvrir le **SQLbuilder**. Sinon, saisissez le code suivant : `Alter table <TableName> auto_increment = 1` directement dans le champ **Query**.

6. Enregistrez le Job en appuyant sur les touches **Ctrl+S** et appuyez sur **F6** pour l'exécuter.

L'auto-incrémentation de la base de données est réinitialisée à 1.

tEXAInput

Propriétés du tEXAInput

Famille de composant	Databases/EXA	
Fonction	Le composant **tEXAInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tEXAInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Host name	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Schema name	Saisissez le nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Saisissez le nom de la table.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Guess schema	Cliquez sur le bouton pour récupérer le schéma de la table.
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données EXA.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Consultez les scénarios associés au composant **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

tEXAOutput

Propriétés du tEXAOutput

Famille de composant	Databases/EXA	
Fonction	Le composant **tEXAOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tEXAOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur.
	Schema name	Saisissez le nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

		Insert : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans

		erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Use commit control*	Cochez cette case afin de définir, dans le champ **Commit every**, qui apparaît alors, le nombre de lignes à traiter avant de commiter.
	Additional JDBC parameters	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tEXAOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données EXA. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Rejects** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	

Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tEXARow

Propriétés du tEXARow

Famille de composant	Databases/EXA	
Fonction	**tEXARow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tEXARow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Schema name	Saisissez le nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

	Table Name	Nom de la table à traiter.
	Query type	La requête peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tEXistConnection

Propriétés du tEXistConnection

Famille de composant	Databases/eXist	
Fonction	Le **tEXistConnection** ouvre une connexion à une base de données **eXist** afin d'effectuer une transaction.	
Objectif	Ouvrir une connexion à une base de données **eXist** afin d'effectuer une transaction.	
Basic settings	*URI*	URI de la base de données à laquelle vous souhaitez vous connecter.
	Collection	Saisissez le chemin d'accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.
	Driver	Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s'affiche. Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.
	Username et Password	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est étroitement lié aux **tEXistGet** et **tEXistPut**. Une fois que vous avez défini les propriétés de connexion du composant, vous pouvez réutiliser cette connexion sans avoir à définir à nouveau les propriétés pour chaque composant **tEXist** utilisé dans le Job. La base de données eXist est un système de gestion de base de données open source, construit à l'aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l'index. Pour plus d'informations concernant XQuery, consultez : http://exist-db.org/exist/apps/doc/documentation.xml (en anglais). Pour plus d'informations concernant les extensions de XQuery Update, consultez : http://exist-db.org/exist/apps/doc/update_ext.xml#	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au **tEXistConnection**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille* du composant **tMysqlConnection**.

tEXistDelete

Propriétés du tEXistDelete

Famille de composant	Databases/eXist	
Fonction	Ce composant supprime des ressources d'une base de données eXist.	
Objectif	Le **tEXistDelete** supprime des ressources spécifiées dans une base de données eXist distante.	
Basic settings	*Use an existing connection/ Component List*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	URI	URI de la base de données à laquelle vous souhaitez vous connecter.
	Collection	Saisissez le chemin d'accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.
	Driver	Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s'affiche. Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.
	Username et Password	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Target Type	Sélectionnez le type de cible : **Resource**(Ressource), **Collection** (Collection), ou **All** (Tous).
	Files	Cliquez sur le bouton **[+]** afin d'ajouter les lignes que vous souhaitez utiliser comme filtres : **Filemask** : saisissez le nom du fichier ou le masque de fichier, en utilisant des caractères de remplacement (*) ou des expressions régulières.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est utilisé en tant que sous-job ou comme composant de sortie. La base de données eXist est un système de gestion de base de données open source, construit à l'aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l'index. Pour plus d'informations concernant XQuery, consultez : http://exist-db.org/exist/apps/doc/documentation.xml (en anglais). Pour plus d'informations concernant les extensions de XQuery Update, consultez : http://exist-db.org/exist/apps/doc/update_ext.xml
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tEXistGet

Propriétés du tEXistGet

Famille de composant	Databases/eXist	
Fonction	Le composant **tEXistGet** récupère des ressources à partir d'un serveur distant de base de données eXist.	
Objectif	Le **tEXistGet** télécharge les ressources sélectionnées à partir d'un serveur distant de base de données dans un répertoire local.	
Basic settings	*Use an existing connection/ Component List*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	URI	URI de la base de données à laquelle vous souhaitez vous connecter.
	Collection	Saisissez le chemin d'accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.
	Driver	Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s'affiche. Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.
	Username et Password	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Local directory	Chemin d'accès au fichier de destination.
	Files	Cliquez sur le bouton **[+]** afin d'ajouter les lignes que vous souhaitez utiliser comme filtres : **Filemask** : saisissez le nom du fichier ou le masque de fichier, en utilisant des caractères de remplacement (*) ou des expressions régulières.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est utilisé en tant que sous-job ou comme composant de sortie. La base de données eXist est un système de gestion de base de données open source, construit à l'aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l'index. Pour plus d'informations concernant XQuery, consultez : XQuery (en anglais). Pour plus d'informations concernant les extensions de XQuery Update, consultez : XQuery update extension
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Récupérer des ressources à partir d'un serveur distant de base de données eXist

Ce scénario décrit un Job à un composant qui permet de récupérer des données à partir d'un serveur de base de données eXist et de télécharger les données dans un répertoire local.

Ce Job simple nécessite un composant : le **tEXistGet**.

1. A partir de la **Palette**, déposez le composant **tEXistGet** dans l'espace de modélisation graphique.

2. Double-cliquez sur le composant **tEXistGet** pour définir ses propriétés dans l'onglet **Basic settings** de sa vue **Component**.

3. Renseignez le champ **URI** en saisissant l'adresse URI de la base de données eXist à laquelle vous souhaitez vous connecter.

   Dans ce scénario, l'URI est *xmldb:exist://192.168.0.165:8080/exist/xmlrpc*. Notez que cette adresse URI est destinée essentiellement à la démonstration et que ce n'est pas une adresse valide.

4. Remplissez le champ **Collection** avec le chemin d'accès à la collection souhaitée sur le serveur de la base de données, */db/talend* dans ce scénario.

5. Renseignez le champ **Driver** en saisissant le pilote de la base de données XML, *org.exist.xmldb.DatabaseImpl* dans ce scénario.

6. Remplissez les champs **Username** et **Password** avec *admin* et *talend* respectivement pour ce scénario.

7. Cliquez sur le bouton **[...]** à côté du champ **Local directory** pour définir le chemin d'accès au répertoire dans lequel le fichier XML téléchargé à partir du serveur de la base de données sera sauvegardé.

   Dans ce scénario, faites pointer ce chemin vers votre bureau, par exemple :

   C:/Documents and Settings/galano/Desktop/ExistGet.

8. Dans le champ **Files**, cliquez sur le bouton **[+]** afin d'ajouter une nouvelle ligne dans la zone **Filemask** et remplissez-la avec un nom de fichier complet pour récupérer les données à partir d'un fichier spécifique sur le serveur, ou avec un masque de fichier (**filemask**) pour récupérer les données à partir d'un ensemble de fichiers. Dans ce scénario, saisissez *dictionary_en.xml*.

9. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

```
<!-- generated by ToXgene Version 1.1a in Wed Jun 23 12:16:00 EDT 2004 -->
<dictionary>
    <e id="E1">
        <hwg>
            <hw>planks</hw>
            <pr>!gzkn+J|i@u</pr>
            <hw>gaul</hw>
            <pr>Zft^OeKY</pr>
            <pos>adv.</pos>
        </hwg>
        <et>
            <cr>E709</cr>
            <cr>E3</cr>
            <cr>E439</cr>
            <cr>E414</cr>
        </et>
        <ss>
            <s>
                <def>forges doubt ironic, sly dugouts:sly, regular patt

</def>
                <qp>
                    <q>
                        <qd>259</qd>
                        <w>closely brave</w>
                        <loc>VbGf~UiSc[</loc>
```

Le fichier XML *dictionary_en.xml* est récupéré et téléchargé dans le répertoire local défini préalablement.

tEXistList

Propriétés du tEXistList

Famille de composant	Databases/eXist	
Fonction	Ce composant liste les ressources stockées sur un serveur distant de base de données.	
Objectif	Le **tEXistList** liste les ressources stockées sur un serveur distant de base de données.	
Basic settings	*Use an existing connection/ Component List*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	URI	URI de la base de données à laquelle vous souhaitez vous connecter.
	Collection	Saisissez le chemin d'accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.
	Driver	Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s'affiche. Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.
	Username et Password	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Files	Cliquez sur le bouton **[+]** afin d'ajouter les lignes que vous souhaitez utiliser comme filtres : **Filemask** : saisissez le nom du fichier ou le masque de fichier, en utilisant des caractères de remplacement (*) ou des expressions régulières.
	Target Type	Sélectionnez le type de cible : **Resource**(Ressource), **Collection** (Collection), ou **All** (Tous).
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_FILE** : nombre de fichiers itérés. Cette variable est une variable **After** et retourne un nombre entier. **CURRENT_FILE** : nom du fichier courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_FILEPATH** : nom du fichier courant ainsi que son chemin d'accès. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé avec un **tEXistGet** afin de récupérer les fichiers listés, par exemple. La base de données eXist est un système de gestion de base de données open source, construit à l'aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l'index. Pour plus d'informations concernant XQuery, consultez : XQuery (en anglais). Pour plus d'informations concernant les extensions de XQuery Update, consultez : http://exist-db.org/exist/apps/doc/update_ext.xml
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé au **tEXistList** , consultez *Scénario : Itération sur un répertoire distant*.

tEXistPut

Propriétés du tEXistPut

Famille de composant	Databases/eXist	
Fonction	Ce composant charge des ressources dans un serveur de base de données.	
Objectif	Le **tEXistPut** charge des fichiers spécifiés d'un répertoire local défini sur un serveur distant de base de données.	
Basic settings	*Use an existing connection/ Component List*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	URI	URI de la base de données à laquelle vous souhaitez vous connecter.
	Collection	Saisissez le chemin d'accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.
	Driver	Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s'affiche. Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.
	Username et Password	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Local directory	Chemin d'accès au fichier de destination.
	Files	Cliquez sur le bouton **[+]** afin d'ajouter les lignes que vous souhaitez utiliser comme filtres : **Filemask** : saisissez le nom du fichier ou le masque de fichier, en utilisant des caractères de remplacement (*) ou des expressions régulières.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est utilisé en tant que sous-job ou comme composant de sortie. La base de données eXist est un système de gestion de base de données open source, construit à l'aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l'index. Pour plus d'informations concernant XQuery, consultez : XQuery (en anglais). Pour plus d'informations concernant les extensions de XQuery Update, consultez : http://exist-db.org/exist/apps/doc/update_ext.xml
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tEXistXQuery

Propriétés du tEXistXQuery

Famille de composant	Databases/eXist	
Fonction	Ce composant utilise des fichiers locaux contenant des requêtes XPath afin d'interroger des fichiers XML stockés dans une base de données distante.	
Objectif	Le **tEXistXQuery** exécute une requête sur des fichiers XML stockés sur une base de données distante, et écrit les résultats dans un fichier XML stocké localement.	
Basic settings	*Use an existing connection/ Component List*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	URI	URI de la base de données à laquelle vous souhaitez vous connecter.
	Collection	Saisissez le chemin d'accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.
	Driver	Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s'affiche. Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.
	Username et Password	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	XQuery Input File	Parcourez votre répertoire jusqu'au fichier local contenant la requête à exécuter.
	Local Output	Parcourez votre répertoire jusqu'à l'emplacement où les résultats de la requête doivent être sauvegardés.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé en tant que composant unique d'un Job mais peut également être utilisé dans un Job plus complexe. La base de données eXist est un système de gestion de base de	

	données open source, construit à l'aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l'index. Pour plus d'informations concernant XQuery, consultez : XQuery (en anglais). Pour plus d'informations concernant les extensions de XQuery Update, consultez : XQuery update extension
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tEXistXUpdate

Propriétés du tEXistXUpdate

Famille de composant	Databases/eXist	
Fonction	Ce composant traite des fichiers XML et met à jour les enregistrements sur un serveur de base de données.	
Objectif	Le **tEXistXUpdate** traite des fichiers XML et met à jour les enregistrements existants sur un serveur de base de données.	
Basic settings	*Use an existing connection/ Component List*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	URI	URI de la base de données à laquelle vous souhaitez vous connecter.
	Collection	Saisissez le chemin d'accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.
	Driver	Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s'affiche. Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.
	Username et Password	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Update File	Parcourez votre répertoire jusqu'au fichier local à utiliser pour mettre à jour les enregistrements de la base de données.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé en tant que composant unique d'un Job mais peut également être utilisé dans un Job plus complexe. La base de données eXist est un système de gestion de base de données open source, construit à l'aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l'index. Pour plus d'informations concernant XQuery, consultez : XQuery (en anglais). Pour plus	

	d'informations concernant les extensions de XQuery Update, consultez : http://exist-db.org/exist/apps/doc/update_ext.xml
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tFirebirdClose

Propriétés du tFirebirdClose

Famille de composant	Databases/Firebird	
Fonction	Le composant **tFirebirdClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Firebird.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tFirebirdConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tFirebirdConnection** et **tFirebirdRollback**. Il est généralement utilisé avec un composant **tFirebirdConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tFirebirdClose**, consultez *tMysqlConnection*.

tFirebirdCommit

Propriétés du tFirebirdCommit

Famille de composant	Databases/Firebird	
Fonction	Le composant **tFirebirdCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tFirebirdConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tFireBirdCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Firebird et notamment avec les composants **tFirebirdConnection** et **tFirebirdRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tFirebirdCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tFirebirdConnection

Propriétés du tFirebirdConnection

Famille de composant	Databases/Firebird	
Fonction	Le composant **tFirebirdConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

		Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec des composants Firebird, notamment les composants **tFirebirdCommit** et **tFirebirdRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé au composant **tFirebirdConnection**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tFirebirdInput

Propriétés du tFirebirdInput

Famille de composant	Databases/ Firebird	
Fonction	Le composant **tFirebirdInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tFirebirdInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Trim all the String/Char columns*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Firebird.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez les scénarios associés au composant générique **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.

- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tFirebirdOutput

Propriétés du tFirebirdOutput

Famille de composant	Databases/ Firebird	
Fonction	Le composant **tFirebirdOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tFirebirdOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.

		Update : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de transaction (cependant pas de rollback) et surtout une meilleure performance des exécutions.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tFirebirdOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Support null in "SQL WHERE" statement	Cochez cette case pour prendre en compte les valeurs Null d'une table de base de données. Assurez-vous que la case **Nullable** est bien cochée pour les colonnes du schéma correspondantes.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Firebird. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tFirebirdRollback

Propriétés du tFirebirdRollback

Famille de composant	Databases/Firebird	
Fonction	Le **tFirebirdRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tFirebirdConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en association avec des composants Firebird, notamment avec le **tFirebirdConnection** et le **tFirebirdCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tFirebirdRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMysqlRollback**.

tFirebirdRow

Propriétés du tFirebirdRow

Famille de composant	Databases/Firebird	
Fonction	**tFirebirdRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tFirebirdRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à traiter.
	Query type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tHiveClose

Le composant **tHiveClose** appartient à deux familles différentes : Big Data et Databases. Pour plus d'informations, consultez *tHiveClose*.

tHiveConnection

Le composant **tHiveConnection** appartient à deux familles différentes : Big Data et Databases. Pour plus d'informations, consultez *tHiveConnection*.

tHiveRow

Le composant **tHiveRow** appartient à deux familles différentes : Big Data et Databases. Pour plus d'informations, consultez *tHiveRow*.

tHSQLDbInput

Propriétés du tHSQLDbInput

Famille de composant	Databases/HSQLDb	
Fonction	Le composant **tHSQLDbInput** lit les données d'une base de données et extrait les champs selon la requête exécutée.	
Objectif	Le **tHSQLDbInput** exécute une requête DB dans un ordre strictement défini devant correspondre à la définition du schéma. Il passe ensuite la liste des champs au composant suivant à l'aide d'un lien **Main row**.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Running Mode	Sélectionnez dans la liste le type de serveur correspondant aux paramètres de votre base de données, parmi les quatre proposés : **HSQLDb Server**, **HSQLDb WebServer**, **HSQLDb In Process Persistent**, **HSQLDb In Memory**.
	Use TLS/SSL sockets	Cochez cette case pour autoriser, si besoin, le mode sécurisé.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database Alias	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	DB path	Spécifiez le chemin d'accès à la base de données à laquelle vous souhaitez vous connecter. Ce champ n'est disponible qu'en mode d'exécution **HSQLDb In Process Persistent**. Par défaut, si la base de données que vous spécifiez dans ce champ n'existe pas, elle sera créée automatiquement. Si vous souhaitez changer la configuration par défaut, modifiez le paramètre de connexion défini dans le champ **Additional JDBC parameters** dans la vue **Advanced settings**.
	Db name	Saisissez le nom de la base de données à laquelle vous souhaitez vous connecter. Ce champ n'est disponible qu'avec les modes d'exécution **HSQLDb In Process Persistent** et **HSQLDb In Memory**.

	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Query type	Peut être **Built-in** ou **Repository**. **Built-in** : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder. **Repository** : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Guess schema	Cliquez sur le bouton pour récupérer le schéma de la table.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Lorsque le mode d'exécution est **HSQLDb In Process Persistent**, vous pouvez configurer le paramètre de connexion `ifexists=true` afin de permettre la connexion à une base de données existante uniquement et éviter de créer une nouvelle base de données.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant couvre toutes les possibilités de requêtes SQL sur une base de données HSQLDb.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate. **Trigger** : Run if, On Component Ok, On Component Error, On Subjob Ok, On Subjob Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate.	

	Trigger : Run if, On Component Ok, On Component Error, On Subjob Ok, On Subjob Error. Pour plus d'informations concernant les connexions, consultez la section relatives aux types de connexions, dans le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE**: Indicates the number of lines processed. This is an After variable and it returns an integer. **QUERY**: Indicates the query to be processed. This is a Flow variable and it returns a string. For further information about variables, see *Studio Talend User Guide*. A Flow variable means it functions during the execution of a component while an After variable means it functions after the execution of a component.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

tHSQLDbOutput

Propriétés du tHSQLDbOutput

Famille de composant	Databases/HSQLDb	
Fonction	Le composant **tHSQLDbOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tHSQLDbOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Running Mode	Sélectionnez dans la liste le type de serveur correspondant aux paramètres de votre base de données, parmi les quatre proposés : **HSQLDb Server**, **HSQLDb WebServer**, **HSQLDb In Process Persistent**, **HSQLDb In Memory**.
	Use TLS/SSL sockets	Cochez cette case pour autoriser, si besoin, le mode sécurisé.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	DB path	Spécifiez le chemin d'accès à la base de données à laquelle vous souhaitez vous connecter. Ce champ n'est disponible qu'en mode d'exécution **HSQLDb In Process Persistent**. Par défaut, si la base de données que vous spécifiez dans ce champ n'existe pas, elle sera créée automatiquement. Si vous souhaitez changer la configuration par défaut, modifiez le paramètre de connexion défini dans le champ **Additional JDBC parameters** dans la vue **Advanced settings**.
	Db name	Saisissez le nom de la base de données à laquelle vous souhaitez vous connecter. Ce champ n'est disponible qu'avec les modes d'exécution **HSQLDb In Process Persistent** et **HSQLDb In Memory**.

	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de

		propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Lorsque le mode d'exécution est **HSQLDb In Process Persistent**, cette propriété supplémentaire est définie comme `ifexists=true` par défaut, ce qui signifie que la base de données sera automatiquement créée lorsque c'est nécessaire. Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tHSQLDbOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données HSQLDb. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate.	

	Trigger : Run if, On Component Ok, On Component Error, On Subjob Ok, On Subjob Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main. **Trigger** : Run if, On Component Ok, On Component Error, On Subjob Ok, On Subjob Error. Pour plus d'informations concernant les connexions, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tHSQLDbRow

Propriétés du tHSQLDbRow

Famille de composant	Databases/HSQLDb	
Fonction	**tHSQLDbRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tHSQLDbRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Running Mode	Sélectionnez dans la liste le type de serveur correspondant aux paramètres de votre base de données, parmi les quatre proposés : **HSQLDb Server**, **HSQLDb WebServer**, **HSQLDb In Process Persistent**, **HSQLDb In Memory**.
	Use TLS/SSL sockets	Cochez cette case pour autoriser, si besoin, le mode sécurisé.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database Alias	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	DB path	Spécifiez le chemin d'accès à la base de données à laquelle vous souhaitez vous connecter. Ce champ n'est disponible qu'en mode d'exécution **HSQLDb In Process Persistent**. Par défaut, si la base de données que vous spécifiez dans ce champ n'existe pas, elle sera créée automatiquement. Si vous souhaitez changer la configuration par défaut, modifiez le paramètre de connexion défini dans le champ **Additional JDBC parameters** dans la vue **Advanced settings**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type	Peut être Built-in ou Repository. **Built-in** : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder. **Repository** : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Lorsque le mode d'exécution est **HSQLDb In Process Persistent**, vous pouvez configurer le paramètre de connexion `ifexists=true` afin de permettre la connexion à une base de données existante uniquement et éviter de créer une nouvelle base de données.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.

Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main,Reject, Iterate. **Trigger** : Run if, On Component Ok, On Component Error, On Subjob Ok, On Subjob Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Iterate. **Trigger** : Run if, On Component Ok, On Component Error, On Subjob Ok, On Subjob Error. Pour plus d'informations concernant les connexions, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.
Global Variables	**QUERY** : indique le nombre de requêtes traitées. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Pour plus d'informations concernant les variables, consultez le Guide utilisateur du *Studio Talend*. Une variable **Flow** signifie qu'elle fonctionne durant l'exécution d'un composant. Une variable **After** signifie qu'elle fonctionne après l'exécution d'un composant.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données*.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL*.

tInterbaseClose

Propriétés du tInterbaseClose

Famille de composant	Databases/Interbase	
Fonction	Le composant **tInterbaseClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données Interbase.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tInterbaseConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tInterbaseConnection** et **tInterbaseRollback**. Ils sont généralement utilisés avec un composant **tInterbaseConnection** car il permet d'ouvrir une connexion pour la transaction en cours.

Pour un scénario associé au composant **tInterbaseClose**, consultez *tMysqlConnection*

tInterbaseCommit

Propriétés du tInterbaseCommit

Famille de composant	Databases/Interbase	
Fonction	Le composant **tInterbaseCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	Sélectionnez le composant **tInterbaseConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tInterbaseCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Interbase et notamment avec les composants **tInterbaseConnection** et **tInterbaseRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tInterbaseCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tInterbaseConnection

Propriétés du tInterbaseConnection

Famille de composant	Databases/Interbase	
Fonction	Le composant **tInterbaseConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

		Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec des composants Interbase, notamment les composants **tInterbaseCommit** et **tInterbaseRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé au composant **tInterbaseConnection**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tInterbaseInput

Propriétés du tInterbaseInput

Famille de composant	Databases/ Interbase	
Fonction	Le composant **tInterbaseInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tInterbaseInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Trim all the String/Char columns*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Interbase.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Consultez les scénarios associés au composant **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tInterbaseOutput

Propriétés du tInterbaseOutput

Famille de composant	Databases/ Interbase	
Fonction	Le composant **tInterbaseOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tInterbaseOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.

		Clear a table : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Clear data in table	Cochez cette case pour que les données soient supprimées de la table sélectionnée avant toute action.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.

		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tInterbaseOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Interbase. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* associé au composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* associé au composant **tMysqlOutput**.

tInterbaseRollback

Propriétés du tInterbaseRollback

Famille de composant	Databases/Interbase	
Fonction	Le **tInterbaseRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tInterbaseConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en association avec des composants Interbase, notamment avec le **tInterbaseConnection** et le **tInterbaseCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tInterbaseRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMySQLRollback**.

tInterbaseRow

Propriétés du tInterbaseRow

Famille de composant	Databases/ Interbase	
Fonction	**tInterbaseRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tInterbaseRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table Name	Nom de la table à traiter.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* associé au composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* associé au composant **tMysqlRow**.

tJavaDBInput

Propriétés du tJavaDBInput

Famille de composant	Databases/JavaDB	
Fonction	Le composant **tJavaDBInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tJavaDBInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Framework	Sélectionnez dans la liste un framework pour votre base de données Java.
	Database	Nom de la base de données.
	DB root path	Parcourez vos fichiers jusqu'à la racine de votre base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type et Query	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Trim all the String/Char columns*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données JavaDB.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez les scénarios du composant **tDBInput**dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tJavaDBOutput

Propriétés du tJavaDBOutput

Famille de composant	Databases/JavaDB	
Fonction	Le composant **tJavaDBOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tJavaDBOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Framework	Sélectionnez dans la liste un framework pour votre base de données Java.
	Database	Nom de la base de données.
	DB root path	Parcourez vos fichiers jusqu'à la racine de votre base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

		Insert : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans

		erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tJavaDBOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données JavaDB. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur	

	le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tJavaDBRow

Propriétés du tJavaDBRow

Famille de composant	Databases/JavaDB	
Fonction	**tJavaDBRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tJavaDBRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Framework Type	Sélectionnez dans la liste un framework pour votre base de données Java.
	Database	Nom de la base de données.
	DB root path	Parcourez vos fichiers jusqu'à la racine de votre base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

	Table Name	Nom de la table à traiter.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tJDBCColumnList

Propriétés du tJDBCColumnList

Famille de composants	Databases/JDBC	
Fonction	Le composant **tJDBCColumList** effectue une opération d'itération dans toutes les colonnes d'une table donnée, grâce à une connexion JDBC définie.	
Objectif	Le **tJDBCColumList** permet de faire une liste des libellés des colonnes d'une table JDBC donnée.	
Basic settings	*Database Type*	Sélectionnez le type de base de données à laquelle vous souhaitez vous connecter.
	Component list	Sélectionnez le composant **tJDBCConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Table name	Nom de la table à lire.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
Global Variables	**CURRENT_COLUMN** : nom de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_COLUMN_TYPE** : ID du type de colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne un nombre entier. **CURRENT_COLUMN_TYPE_NAME** : nom du type de colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_COLUMN_PRECISION** : précision de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne un nombre entier. **CURRENT_COLUMN_SCALE** : échelle de la colonne sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne un nombre entier. **NB_COLUMN** : nombre de colonnes itérées jusqu'à présent. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Il faut utiliser ce composant en association avec les autres composants JDBC, notamment avec le **tJDBCConnection**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Itérer une table de base de données et lister le nom des colonnes de la table*.

tJDBCClose

Propriétés du tJDBCClose

Famille de composant	Databases/JDBC	
Fonction	Le composant **tJDBCClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données JDBC.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tJDBCConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tJDBCConnection** et **tJDBCRollback**. Il est généralement utilisé avec un composant **tJDBCConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tJDBCClose**, consultez *tMysqlConnection*

tJDBCCommit

Propriétés du tJDBCCommit

Famille de composant	Databases/JDBC	
Fonction	Le composant **tJDBCCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	Sélectionnez le composant **tJDBCConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tJDBCCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants JDBC et notamment avec les composants **tJDBCConnection** et **tJDBCRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tJDBCCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tJDBCConnection

Propriétés du tJDBCConnection

Famille de composant	Databases/JDBC	
Fonction	Le composant **tJDBCConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*JDBC URL*	Saisissez l'URL JDBC dans le champ associé pour vous connecter à la base de données. Par exemple, entrez *jdbc:mysql://IP address/database name* pour vous connecter à une base de données MySQL.
	Driver JAR	Cliquez sur le bouton **[+]** sous le tableau afin d'ajouter autant de lignes que nécessaire pour charger différents JARs. Dans chaque ligne, cliquez ensuite sur le bouton **[...]** pour ouvrir l'assistant **[Select Module]**, dans lequel vous pouvez sélectionner le pilote JAR de votre choix pour chaque ligne.
	Driver Class	Renseignez la classe du driver dans le champ **Driver Class** en fonction de votre connexion. Par exemple, entrez *com.mysql.jdbc.Driver* dans ce champ pour vous connecter à une base de données MySQL.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*.
	Auto Commit	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut,

		la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
Utilisation	Ce composant est généralement utilisé avec des composants JDBC et notamment avec les composants **tJDBCCommit** et **tJDBCRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tJDBCConnection**, consultez *tMysqlConnection*.

tJDBCInput

Propriétés du tJDBCInput

Famille de composant	Databases/JDBC	
Fonction	Le composant **tJDBCInput** lit toute base de données via une API JDBC et en extrait des champs à l'aide de requêtes. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job **Talend** Map/Reduce afin de générer du code Map/Reduce. Dans le cas présent, le composant **tJDBCInput** appartient à la famille de composants Map/Reduce. Pour plus d'informations, consultez *Le composant tJDBCInput dans les Jobs Talend Map/Reduce*.	
Objectif	Le **tJDBCInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	JDBC URL	Entrez le chemin d'accès à la base de données.
	Driver JAR	Cliquez sur le bouton **[+]** sous le tableau afin d'ajouter autant de lignes que nécessaire pour charger différents JAR. Dans chaque ligne, cliquez ensuite sur le bouton **[...]** pour ouvrir l'assistant **[Select Module]**, dans lequel vous pouvez sélectionner le pilote JAR de votre choix pour chaque ligne.
	Class Name	Dans le champ **Class name**, renseignez le nom de la classe pour qu'elle soit indiquée dans le driver.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.

		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à lire.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. *Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l'alias de la source de données spécifié.* Cette option est indisponible lorsque la case **Use an existing connection** est cochée.
Advanced settings	*Use cursor*	Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d'optimiser les performances.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	Enable Mapping File for Dynamic	Cochez cette case pour utiliser le fichier de mappage de métadonnées spécifié lors de la lecture de données d'une colonne de type dynamique. Cette case est décochée par défaut. Pour plus d'informations concernant les fichiers de mappage de métadonnées, consultez la section sur la conversion de type du *Guide Utilisateur duStudio Talend*.
	Mapping File	Spécifiez le fichier de mappage de métadonnées à utiliser en sélectionnant un type de base de données dans la liste. Ce champ apparaît uniquement lorsque la case **Enable Mapping File for Dynamic** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données utilisant une connexion JDBC.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Le composant tJDBCInput dans les Jobs Talend Map/Reduce

*Les informations contenues dans cette section ne s'appliquent que pour les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend**. Elles ne s'appliquent pas aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job **Talend** Map/Reduce, le composant **tJDBCInput**, ainsi que les autres composants Map/Reduce qui le précédent, génèrent du code Map/Reduce natif. Cette section présente les propriétés spécifiques du **tJDBCInput** lorsqu'il est utilisé dans le cas présent. Pour plus d'informations sur les Jobs **Talend** Map/Reduce Job, consultez le *Guide de prise en main*.

Famille de composant	MapReduce/Input	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**
		Built-in : Propriétés utilisées ponctuellement.
		Repository: Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Bsaic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du* Studio Talend.
	JDBC URL	Entrez le chemin d'accès à la base de données. Par exemple, si la base de données MySQL s'appelle *Talend* et est hébergée sur une machine dont l'adresse IP est *XX.XX.XX.XX* et le port est *3306*, l'URL doit être *jdbc:mysql://XX.XX.XX.XX:3306/Talend*.
	Driver JAR	Cliquez sur le bouton **[+]** situé sous le tableau afin d'ajouter autant de lignes que nécessaire pour charger différents JAR. Dans chaque ligne, cliquez ensuite sur le bouton **[...]** pour ouvrir l'assistant **[Select Module]** dans lequel vous pouvez sélectionner le pilote JAR de votre choix pour chaque ligne.
	Class Name	Dans le champ **Class name**, renseignez le nom de la classe pour qu'elle soit indiquée dans le driver. Par exemple, pour le pilote *mysql-connector-java-5.1.2.jar*, le nom à saisir est *org.gjt.mm.mysql.Driver*.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données.

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Table Name	Saisissez le nom de la table à lire.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Utilisation	Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme composant de début et requiert un composant de transformation comme lien de sortie. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.	
Limitation	Il est recommandé d'utiliser les bases de données suivantes avec la version Map/Reduce du composant : DB2, Informix, MSSQL, MySQL, Netezza, Oracle, Postgres, Teradata et Vertica. Ce composant peut également fonctionner avec d'autres bases de données. Cependant, ces dernières n'ont pas nécessairement été testées.	

Scénarios associés

Pour un scénario associé, consultez les scénarios du composant **tDBInput** dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tJDBCOutput

Propriétés du tJDBCOutput

Famille de composant	Databases/ JDBC	
Fonction	Le composant **tJDBCOutput** écrit, met à jour, modifie ou supprime les données d'une base de données. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job **Talend** Map/Reduce ou Storm : afin de générer du code Map/Reduce. • Dans un Job Map/Reduce, ce composant génère du code MapReduce et peut écrire des données uniquement dans une base de données.Pour plus d'informations, consultez *Le composant tJDBCOutput dans les Jobs Talend Map/Reduce*. • Dans un Job Storm, ce composant génère du code Storm. Pour plus d'informations, consultez *Le tJDBCOutput dans des Jobs Storm Talend*.	
Objectif	Le **tJDBCOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	JDBC URL	Entrez le chemin d'accès à la base de données.
	Driver JAR	Cliquez sur le bouton **[+]** sous le tableau afin d'ajouter autant de lignes que nécessaire pour charger différents JAR. Dans chaque ligne, cliquez ensuite sur le bouton **[...]** pour ouvrir l'assistant **[Select Module]**, dans lequel vous pouvez sélectionner le pilote JAR de votre choix pour chaque ligne.
	Class Name	Dans le champ **Class name**, renseignez le nom de la classe pour qu'elle soit indiquée dans le driver.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.

	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. *Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l'alias de la source de données spécifié.* Cette option est indisponible lorsque la case **Use an existing connection** est cochée.

Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tJDBCOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	Use Batch	Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.
	Batch Size	Spécifiez le nombre d'enregistrements à traiter dans chaque lot. La valeur par défaut de ce champ est *10000*. Ce champ est disponible uniquement lorsque la case **Use Batch** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données JDBC. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour	

	un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Le composant tJDBCOutput dans les Jobs Talend Map/Reduce

*Les informations contenues dans cette section ne s'appliquent que pour les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend**. Elles ne s'appliquent pas aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job **Talend** Map/Reduce, le composant **tJDBCOutput**, ainsi que les autres composants Map/Reduce qui le précédent, génèrent du code Map/Reduce natif. Cette section présente les propriétés spécifiques du **tJDBCInput** lorsqu'il est utilisé dans le cas présent. Pour plus d'informations sur les Jobs **Talend** Map/Reduce Job, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	MapReduce/Output	
Fonction	Le composant **tJDBCOutput** écrit les données d'une base de données connectée à une API JDBC.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du* Studio Talend.
	JDBC URL	Entrez le chemin d'accès à la base de données. Par exemple, si la base de données MySQL s'appelle *Talend* et est hébergée sur une machine dont l'adresse IP est *XX.XX.XX.XX* et le port est *3306*, l'URL doit être *jdbc:mysql://XX.XX.XX.XX:3306/Talend*.
	Driver JAR	Cliquez sur le bouton **[+]** situé sous le tableau afin d'ajouter autant de lignes que nécessaire pour charger différents JAR. Dans chaque ligne, cliquez ensuite sur le bouton **[...]** pour ouvrir l'assistant **[Select Module]** dans lequel vous pouvez sélectionner le pilote JAR de votre choix pour chaque ligne.
	Class Name	Dans le champ **Class name**, renseignez le nom de la classe pour qu'elle soit indiquée dans le driver. Par exemple, pour le pilote *mysql-connector-java-5.1.2.jar*, le nom à saisir est *org.gjt.mm.mysql.Driver*.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table name	Nom de la table à écrire. Notez que cette table doit existe et qu'une seule table ne peut être écrite à la fois.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.

		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*Use Batch Size*	Nombre de lignes dans chaque lot de traitement de données.
Utilisation	Dans un Job **Talend** Map/Reduce, ce composant est utilisé en tant que composant de fin et requiert un composant de transformation comme lien d'entrée. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Une fois que le Job Map/Reduce est ouvert dans l'espace de modélisation, le **tJDBCOutput** ainsi que la famille de composant apparaît dans la **Palette** du *Studio Talend*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.	
Limitation	Il est recommandé d'utiliser les bases de données suivantes avec la version Map/Reduce du composant : DB2, Informix, MSSQL, MySQL, Netezza, Oracle, Postgres, Teradata et Vertica. Ce composant peut également fonctionner avec d'autres bases de données. Cependant, ces dernières n'ont pas nécessairement été testées.	

Le tJDBCOutput dans des Jobs Storm Talend

*Les informations contenues dans cette section ne s'appliquent que pour les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend**. Elles ne s'appliquent pas aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job Storm **Talend**, le **tJDBCOutput**, ainsi que les autres composants Storm le précédent, génèrent nativement du code Storm pour construire un Topology Storm à utiliser dans votre écosystème Storm. Cette section présente les propriétés spécifiques du **tJDBCOutput** dans cette situation. Pour plus d'informations concernant les Jobs Storm **Talend**, consultez *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	Storm/Output	
Fonction	Le **tJDBCOutput** écrit des entrées de tout type dans la base de données connectée à une API JDBC.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du* Studio Talend.
	JDBC URL	Entrez le chemin d'accès à la base de données. Par exemple, si la base de données MySQL s'appelle *Talend* et est hébergée sur une machine dont l'adresse IP est *XX.XX.XX.XX* et le port est *3306*, l'URL doit être *jdbc:mysql://XX.XX.XX.XX:3306/Talend*.

	Driver JAR	Cliquez sur le bouton **[+]** situé sous le tableau afin d'ajouter autant de lignes que nécessaire pour charger différents JAR. Dans chaque ligne, cliquez ensuite sur le bouton **[...]** pour ouvrir l'assistant **[Select Module]** dans lequel vous pouvez sélectionner le pilote JAR de votre choix pour chaque ligne.
	Class Name	Saisissez le nom de la Classe vers laquelle pointer dans le pilote. Par exemple, pour le pilote *mysql-connector-java-5.1.2.jar*, le nom à saisir est *org.gjt.mm.mysql.Driver*.
	Username et *Password*	Données d'authentification de l'utilisateur à la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table name	Nom de la table à écrire. Notez qu'elle doit exister et qu'une seule table peut être écrite à la fois.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
Utilisation	Dans un Job Storm **Talend**, ce composant est utilisé comme composant de fin. Les autres composants utilisés avec celui-ci doivent également être des composants Storm. Ils génèrent nativement du code Storm pouvant être exécuté directement dans un système Storm. Ce composant et la famille Storm à laquelle il appartient sont disponibles uniquement lorsque vous créez un Job Storm. Notez que, dans cette documentation, sauf lorsqu'explicitement indiqué, un scénario présente uniquement des Jobs de type **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données.	
Storm Connection	Vous devez utiliser l'onglet **Storm Configuration** dans la vue **Run** pour définir la connexion à un système Storm donné pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.	
Limitation	n/a	

Scénarios associés

Pour un scénario associé au **tJDBCOutput**, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez également consulter un Job **Talend** Map/Reduce utilisant une version Map/Reduce du **tJDBCOutput** :

- *Scénario 2 : Dédoublonner des entrées en utilisant des composants Map/Reduce*

tJDBCRollback

Propriétés du tJDBCRollback

Famille de composant	Databases/JDBC	
Fonction	Le **tJDBCRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tJDBCConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en association avec des composants JDBC, notamment avec le **tJDBCConnection** et le **tJDBCCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tJDBCRollback**, consultez *tMysqlConnection*.

tJDBCRow

Propriétés du tJDBCRow

Famille de composant	Databases/JDBC	
Fonction	**tJDBCRow** est le composant générique pour tout type de base de données utilisant une API JDBC. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tJDBCRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	JDBC URL	Entrez le chemin d'accès à la base de données.
	Driver JAR	Cliquez sur le bouton **[+]** sous le tableau afin d'ajouter autant de lignes que nécessaire pour charger différents JARs. Dans chaque ligne, cliquez ensuite sur le bouton **[...]** pour ouvrir l'assistant **[Select Module]**, dans lequel vous pouvez sélectionner le pilote JAR de votre choix pour chaque ligne.
	Class Name	Renseignez le nom de la classe pour qu'elle soit indiquée dans le driver.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

	Table Name	Nom de la table à traiter.
	Query type	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. *Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l'alias de la source de données spécifié.* Cette option est indisponible lorsque la case **Use an existing connection** est cochée.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement	

	dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur toute base de données utilisant une connexion JDBC et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tJDBCSP

Propriétés du tJDBCSP

Famille de composant	Databases/JDBC	
Fonction	Le composant **tJDBCSP** appelle une procédure stockée de base de données.	
Objectif	Le **tJDBCSP** permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.	
Basic settings	*JDBC URL*	Entrez le chemin d'accès à la base de données.
	Driver JAR	Cliquez sur le bouton **[+]** sous le tableau afin d'ajouter autant de lignes que nécessaire pour charger différents JARs. Dans chaque ligne, cliquez ensuite sur le bouton **[...]** pour ouvrir l'assistant **[Select Module]**, dans lequel vous pouvez sélectionner le pilote JAR de votre choix pour chaque ligne.
	Class Name	Dans le champ **Class name**, renseignez le nom de la classe pour qu'elle soit indiquée dans le driver.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Dans une procédure stockée, le schéma est un paramètre d'entrée. Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	SP Name	Saisissez le nom exact de la Procédure Stockée (SP).
	Is Function / *Return result in*	Cochez cette case si une seule valeur doit être retournée.

		Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.
	Parameters	Cliquez sur le bouton **[+]** et sélectionnez dans le champ **Schema Columns** les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure. Sélectionnez le **Type** de paramètre : **IN** : paramètre d'entrée (Input) **OUT** : paramètre de sortie (Output)/valeur retournée **IN OUT** : les paramètres d'entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction). **RECORDSET** : les paramètres d'entrée doivent être retournées sous forme d'ensemble de valeurs, au lieu d'une valeur unique. Consultez le composant *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.
	Specify a data source alias	Cochez cette case et spécifiez l'alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. *Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l'alias de la source de données spécifié.* Cette option est indisponible lorsque la case **Use an existing connection** est cochée.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d'entrée sont autorisés.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données.	

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Exécuter une procédure stockée dans le hub MDM* du composant **tMysqlSP**.
- *Scénario : Vérifier le format de numéros à l'aide d'une procédure stockée* du composant **tOracleSP**.

Consultez également *Scénario 1 : Insérer des données dans des tables mère/fille* si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.

tJDBCTableList

Propriétés du tJDBCTableList

Famille de composant	Databases/JDBC	
Fonction	Le composant **tJDBCTableList** effectue une opération d'itération sur toutes les tables d'une base de données, grâce à une connexion JDBC définie.	
Objectif	Le composant **tJDBCTableList** permet de lister les noms des tables JDBC grâce aux commandes SELECT et WHERE.	
Basic settings	*Database type*	Sélectionnez dans la liste la base de données que vous utilisez.
	Component list	Sélectionnez le composant de connexion **tJDBCConnection** ou le composant de connexion de la base de données sélectionnée dans la liste **Database type**.
Advanced settings	*Use filter*	Cochez cette case afin de filtrer les tables sur lesquelles effectuer une itération.
	Regular expression for tables name	Saisissez l'expression régulière permettant d'identifier le nom des tables. Ce champ est disponible lorsque la case **Use filter** est cochée.
	Filter criteria	Sélectionnez le critère permettant de filtrer le nom des tables. **Include** : effectue une itération sur les noms de tables identifiés par l'expression régulière uniquement. **Exclude** : effectue une itération sur les noms de tables non identifiés par l'expression régulière. Cette liste est disponible lorsque la case **Use filter** est cochée.
	tStatCatcher statistics	Cochez cette case afin de collecter les données de log au niveau des composants.
Global Variables	**CURRENT_TABLE** : nom de la table sur laquelle se fait l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_TABLE** : nombre de tables itérées jusqu'à présent. Cette variable est une variable **Flow** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé en association avec les autres composants JDBC, notamment avec le **tJDBCConnection**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Itérer une table de base de données et lister le nom des colonnes de la table*.

tLDAPAttributesInput

Propriétés du tLDAPAttributesInput

Famille de composant	Databases/LDAP	
Fonction	Le composant **tLDAPAttributesInput** analyse chaque élément trouvé via une requête LDAP et liste une série d'attributs associés à l'objet.	
Objectif	Le **tLDAPAttributesInput** exécute une LDAP basée sur un filtre prédéfini et correspondant au schéma défini. Puis la liste de champ est transmise au composant suivant via une connexion **Main row**.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et cliquez sur le composant **tLDAPConnection** dans la liste **Component List**, afin de réutiliser les informations de la connexion que vous avez précédemment définie.
	Host	Adresse IP du serveur d'annuaire LDAP.
	Port	Numéro du port d'écoute du serveur.
	Base DN	Chemin d'accès à l'arborescence de l'utilisateur autorisé.
	Protocol	Sélectionnez le type de protocole dans la liste. **LDAP** : aucun codage n'est utilisé. **LDAPS** : LDAP sécurisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît. Une fois sélectionné, le mode avancé vous permet de spécifier le répertoire et le mot de passe de la clé secrète du certificat pour le stockage d'un CA spécifique. Cependant, vous pouvez toujours désactiver la validation du certificat, en cochant la case **Trust all certs**. **TLS** : le certificat est utilisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît et est utilisée de la même façon que pour le type **LPDAPS**.
	Authentication User et *Password*	Cochez **Authentication** si une connexion LDAP est nécessaire. Notez que le login doit correspondre à la syntaxe LDAP pour être valide. Par exemple : "cn=Directory Manager". Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Filter	Saisissez le filtre comme attendu par l'annuaire LDAP db.
	Multi valued field separator	Saisissez le séparateur de valeur des champs à valeurs multiples.
	Alias dereferencing	Sélectionnez l'option dans la liste. **Never** permet l'amélioration des performances de recherche si vous êtes sûr qu'aucun alias n'est déréférencé. Par défaut, utilisez **Always** : **Always** : les alias sont toujours déréférencés. **Never** : les alias ne sont jamais déréférencés.

		Searching : cette option déréférence les alias uniquement après la résolution du nom. **Finding** : cette option déréférence les alias uniquement lors de la résolution du nom.
	Referral handling	Sélectionnez l'option dans la liste : **Ignore** : ne tient pas compte des redirections des requêtes **Follow** : tient compte des redirections des requêtes
	Limit	Cette option permet de limiter le nombre d'enregistrement lus, si nécessaire.
	Time Limit	Cette option permet de limiter la durée de connexion à l'annuaire.
	Paging	Spécifiez le nombre d'entrées retournées en une fois par le serveur LDAP.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. *Comme ce composant sert à lister les attributs associés à un élément LDAP, sont schéma est prédéfini. Conservez ces colonnes définies, même si vous avez besoin d'ajouter de nouvelles colonnes. Pour cela, utilisez le mode Built-In.*
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Le schéma prédéfini comprend : - *objectclass* : liste des classes d'éléments - *mandatoryattributes* : liste des attributs obligatoires pour ces classes - *optionalattributes* : liste des attributs facultatifs pour ces classes - *objectattributes* : liste des attributs essentiels aux éléments analysés.
Advanced settings	*Class Definition Root*	Spécifiez la racine de l'espace de nom de l'élément Class definition.

	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **RESULT_NAME** : nom de l'entrée LDAP correspondant au filtre de recherche. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les requêtes LDAP possibles. Note : Appuyez sur **Ctrl + Espace** pour accéder à la liste des variables globales, notamment la variable **GetResultName** permettant de récupérer automatiquement la base correspondante.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Le composant **tLDAPAttributesInput** fonctionne de manière similaire au **tLDAPInput**. Vous pouvez donc consulter *Scénario : Afficher le contenu filtré d'un annuaire LDAP*.

tLDAPClose

Propriétés du tLDAPClose

Famille de composant	Databases/LDAP	
Fonction	Le composant **tLDAPClose** ferme la connexion au serveur d'un annuaire connecté.	
Objectif	Le **tLDAPClose** est utilisé pour arrêter une connexion à un serveur d'un annuaire afin de libérer des ressources occupées.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tLDAPConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau des composants.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants LDAP, notamment avec le **tLDAPConnection**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tLDAPConnection

Propriétés du tLDAPConnection

Famille de composant	Databases/LDAP	
Fonction	Le composant **tLDAPConnection** ouvre une connexion au serveur d'un annuaire pour effectuer des transactions de données.	
Objectif	Le **tLDAPConnection** crée une connexion au serveur d'un annuaire LDAP. Cette connexion peut être invoquée par d'autres composants devant accéder à l'annuaire LDAP, c'est-à-dire les composants **tLDAPInput**, **tLDAPOutput**, etc.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur d'annuaire LDAP.
	Port	Numéro du port d'écoute du serveur.
	Protocol	Sélectionnez le type de protocole dans la liste. **LDAP** : aucun codage n'est utilisé **LDAPS** : LDAP sécurisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît. Une fois sélectionné, le mode avancé vous permet de spécifier le répertoire et le mot de passe de la clé secrète du certificat pour le stockage d'un CA spécifique. Cependant, vous pouvez toujours désactiver la validation du certificat, en cochant la case **Trust all certs**. **TLS** : le certificat est utilisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît et est utilisée de la même façon que pour le type **LPDAPS**.
	Base DN	Chemin d'accès à l'arborescence de l'utilisateur autorisé.
	User et *Password*	Renseignez les champs **User** et **Password** comme requis par l'annuaire. Notez que le login doit correspondre à la syntaxe LDAP pour être valide. Par exemple : "cn=Directory Manager". Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Alias dereferencing	Sélectionnez l'option dans la liste. **Never** permet l'amélioration des performances de recherche si vous êtes sûr qu'aucun alias n'est déréférencé. Par défaut, utilisez **Always** : **Always** : les alias sont toujours déréférencés. **Never** : les alias ne sont jamais déréférencés. **Searching** : cette option déréférence les alias uniquement après la résolution du nom. **Finding** : cette option déréférence les alias uniquement lors de la résolution du nom.

	Referral handling	Sélectionnez l'option dans la liste : **Ignore** : ne tient pas compte des redirections des requêtes. **Follow** : tient compte des redirections des requêtes.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant doit être utilisé avec d'autres composants LDAP, notamment les composants **tLDAPInput** et **tLDAPOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié au **tLDAPInput** et au **tLDAPOutput**, puisqu'il vous permet de ne pas avoir à toujours saisir les informations de connexion lors de l'utilisation de différents **tLDAPInput** ou **tLDAPOutput**.

Pour un scénario associé au **tLDAPConnection**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tLDAPInput

Propriétés du tLDAPInput

Famille de composant	Databases/LDAP	
Fonction	Le composant **tLDAPInput** lit un répertoire et extrait les données en fonction d'un filtre défini.	
Objectif	Le **tLDAPInput** exécute une LDAP basée sur un filtre prédéfini et correspondant au schéma défini. Puis la liste de champ est transmise au composant suivant via une connexion **Main row**.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur de Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Host	Adresse IP du serveur d'annuaire LDAP.
	Port	Numéro du port d'écoute du serveur.
	Base DN	Chemin d'accès à l'arborescence de l'utilisateur autorisé. Pour retrouver les informations complètes du Distinguished Name, ajoutez un champ appelé *DN* dans le schéma, en minuscules ou en majuscules.
	Protocol	Sélectionnez le type de protocole dans la liste. **LDAP** : aucun codage n'est utilisé **LDAPS** : LDAP sécurisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît. Une fois sélectionné, le mode avancé vous permet de spécifier le répertoire et le mot de de passe de la clé secrète du certificat pour le stockage d'un CA spécifique. Cependant, vous pouvez toujours désactiver la validation du certificat, en cochant la case **Trust all certs**. **TLS** : le certificat est utilisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît et est utilisée de la même façon que pour le type **LPDAPS**.
	Authentication User et *Password*	Cochez **Authentication** si une connexion LDAP est nécessaire. Notez que le login doit correspondre à la syntaxe LDAP pour être valide. Par exemple : "cn=Directory Manager".

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Filter	Saisissez le filtre comme attendu par l'annuaire LDAP db.
	Multi valued field separator	Saisissez le séparateur de valeur des champs à valeurs multiples.
	Alias dereferencing	Sélectionnez l'option dans la liste. **Never** permet l'amélioration des performances de recherche si vous êtes sûr qu'aucun alias n'est déréférencé. Par défaut, utilisez **Always** : **Always** : les alias sont toujours déréférencés. **Never** : les alias ne sont jamais déréférencés. **Searching** : cette option déréférence les alias uniquement après la résolution du nom. **Finding** : cette option déréférence les alias uniquement lors de la résolution du nom.
	Referral handling	Sélectionnez l'option dans la liste : **Ignore** : ne tient pas compte des redirections des requêtes. **Follow** : tient compte des redirections des requêtes.
	Limit	Cette option permet de limiter le nombre d'enregistrement lus, si nécessaire.
	Time Limit	Cette option permet de limiter la durée de connexion à l'annuaire.
	Paging	Spécifiez le nombre d'entrées retournées en une fois par le serveur LDAP.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. *Seuls trois types de données sont supportés : String, byte[] et List. Vous pouvez utiliser un **tMap** pour effectuer une conversion, si nécessaire.*
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **RESULT_NAME** : nom de l'entrée LDAP correspondant au filtre de recherche. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les requêtes LDAP possibles. Note : Appuyez sur **Ctrl + Espace** pour accéder à la liste des variables globales, inclus la variable **GetResultName** permettant de récupérer automatiquement la base correspondante.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Afficher le contenu filtré d'un annuaire LDAP

Le Job décrit ci-dessous filtre simplement l'annuaire LDAP et affiche le résultat dans la console de la vue **Run**.

1. Cliquez et déposez les composants **tLDAPInput** et **tLogRow**.
2. Paramétrez les propriétés du **tLDAPInput**.
3. Sélectionnez **Repository** dans le champ **Property type** si vous avez stocké les informations de connexion LDAP dans le **Metadata Manager** du **Repository**. Puis sélectionnez l'entrée adéquate dans la liste.
4. En mode **Built-In**, renseignez manuellement les champs **Host** et **Port**. Host peut être l'adresse IP du serveur de l'annuaire LDAP ou son nom DNS.
5. Aucune **Base DN** particulière ne doit être paramétrée.

6. Puis dans le champ **Protocol**, sélectionnez le protocole adéquat dans la liste. Dans cet exemple : un protocole **LDAP** simple est utilisé.

7. Cochez la case **Authentication** et renseignez les informations de connexion pour lire l'annuaire, si nécessaire. Dans ce scénario, aucune authentification n'est nécessaire.

8. Dans la zone **Filter**, saisissez la commande sur laquelle est basée la sélection de données. Dans cet exemple, le filtre est : `(&(objectClass=inetorgperson)&(uid=PIERRE DUPONT))`.

9. Renseignez le champ **Multi-valued field separator** avec une virgule, puisque certains champs contiennent plus d'une valeur séparée par une virgule.

10. Comme vous ne savez pas si des alias sont utilisés dans l'annuaire LDAP, sélectionnez **Always** dans la liste.

11. Sélectionnez **Ignore** dans le champ **Referral handling**.

12. Dans le champ **Limit**, définissez la limite à **100** dans cet exemple.

Column	Db Column	Key	Type	Nullable	Date P...	Length	Pre...
dc	dc	☐	String	☑		255	
ou	ou	☐	String	☑		255	
objectClass	objectClass	☐	String	☑		255	
mail	mail	☐	String	☑		255	
uid	uid	☐	String	☑		255	
dn	dn	☐	String	☐			

13. Définissez le **Schéma** tel qu'il est requis par l'annuaire LDAP. Dans cet exemple, le schéma est composé de six colonnes dont les colonnes `objectClass` et `uid` qui seront filtrées.

14. Dans le composant **tLogRow**, aucune configuration particulière n'est exigée.

```
Starting job testLDAPInput at 18:05 18/09/2007.
|DATA|top,person,organizationalPerson,inetorgperson,a4400user|mhirt78@talend.com|PIERRE DUPONT|
Job testLDAPInput ended at 18:05 18/09/2007. [exit code=0]
```

Seule une entrée de l'annuaire correspond aux critères du filtre défini dans le composant **tLDAPInput**.

tLDAPOutput

Propriétés du tLDAPOutput

Famille de composant	Databases/LDAP	
Fonction	Le composant **tLDAPOutput** écrit dans un annuaire LDAP.	
Objectif	Le **tLDAPOutput** exécute une requête LDAP basée sur un filtre prédéfini et correspondant à un schéma défini. Puis la liste de champ est transmise au composant suivant via une connexion **Main row**.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Host	Adresse IP du serveur d'annuaire LDAP.
	Port	Numéro de port d'écoute du serveur.
	Base DN	Chemin d'accès à l'arborescence de l'utilisateur autorisé.
	Protocol	Sélectionnez le type de protocole dans la liste. **LDAP** : aucun codage n'est utilisé **LDAPS** : LDAP sécurisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît. Une fois sélectionné, le mode avancé vous permet de spécifier le répertoire et le mot de passe de la clé secrète du certificat pour le stockage d'un CA spécifique. Cependant, vous pouvez toujours désactiver la validation du certificat, en cochant la case **Trust all certs**. **TLS** : le certificat est utilisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît et est utilisée de la même façon que pour le type **LPDAPS**.
	User et *Password*	Renseignez les champs **User** (utilisateur) et **Password** (mot de passe) conformément à l'annuaire. Notez que le login doit correspondre à la syntaxe LDAP pour être valide. Par exemple : "cn=Directory Manager".

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Multi valued field separator	Saisissez le séparateur de valeurs dans les champs multi-valués.
	Alias dereferencing	Sélectionnez l'option dans la liste. **Never** permet l'amélioration des performances de recherche si vous êtes sûr qu'aucun alias n'est déréférencé. Par défaut, utilisez **Always**. **Always** : les alias sont toujours déréférencés. **Never** : les alias ne sont jamais déréférencés. **Searching** : cette option déréférence les alias uniquement après la résolution du nom. **Finding** : cette option déréférence les alias uniquement lors de la résolution du nom.
	Referral handling	Sélectionnez l'option dans la liste : **Ignore** : ne tient pas compte des redirections des requêtes. **Follow** : tient compte des redirections des requêtes.
	Insert mode	Sélectionnez l'action dans la liste : **Add** : ajoute une valeur dans un attribut multi-valué **Insert** : insère de nouvelles données **Update** : met à jour les données existantes **Delete** : supprime les données sélectionnées de l'annuaire **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. *Lorsque l'action **Update** est sélectionnée, allez dans la vue **Advanced settings** et cochez la case **Use Attribute Options (for update mode)** pour afficher la table **Attribute Options**. Sélectionnez l'attribut à mettre à jour dans la colonne **Attribute name** et sélectionnez **Replace** dans la colonne **Option**.*
	DN Column Name	Sélectionnez, à partir de la liste déroulante, le type d'entrée LDAP utilisé.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**.
Advanced settings	*Use Attribute Options (for update mode)*	Cochez cette case afin de sélectionner l'attribut souhaité (y compris **dn**, **dc**, **ou**, **objectClass**, **mail** et **uid**) et l'opération correspondante (notamment **Add**, **Replace**, **Remove Attribute**, et **Remove Value**)
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les requêtes LDAP possibles. Note : Appuyez sur **Ctrl + Espace** pour accéder à la liste des variables globales, inclus la variable **GetResultName** permettant de récupérer automatiquement la base correspondante. Ce composant permet de faire des actions sur une table ou les données d'une table d'une base de données. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Editer des données dans un annuaire LDAP

Le scénario suivant décrit un Job qui lit un annuaire LDAP, met à jour l'e-mail d'une entrée sélectionnée et affiche la sortie avant de mettre à jour l'annuaire LDAP en sortie. C'est-à-dire qu'aucun alias n'est déréférencé et qu'aucune requête n'est redirigée. Ce scénario est basé sur le scénario du **LDAPInput** (cf. *Scénario : Afficher le contenu filtré d'un annuaire LDAP*). Le résultat obtenu est une entrée simple, correspondant à une personne organisationnelle dont l'e-mail est mis à jour.

La sortie présente les champs suivants : *dn*, *uid* et *mail*, comme défini dans le Job.

Déposer les composants

1. Cliquez et déposez les composants **tLDAPInput**, **tLDAPOutput**, **tMap** et **tLogRow** dans l'espace de modélisation.

2. Connectez le composant **tLDAPInput** au **tMap** à l'aide d'un lien **Row** > **Main**.

3. Reliez le **tMap** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

4. Reliez le **tLogRow** au **tLDAPOutput** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Dans la vue **Component** du **tLDAPInput**, paramétrez les informations de connexion au serveur d'annuaire LDAP, ainsi que les filtres tel que décrit dans *Scénario : Afficher le contenu filtré d'un annuaire LDAP*.

   Simplifiez le schéma en enlevant les champs suivants : *dc*, *ou*, *objectclass*.

2. Ouvrez le mapper pour définir les changements à exécuter.

   Glissez et déposez la colonne *uid* de la table d'entrée (input) à la table de sortie (output) puisque aucune modification n'est nécessaire pour cette colonne.

3. Dans le champ **Expression** de la colonne *dn* de la table de sortie (output), saisissez l'expression exacte attendue par le serveur LDAP afin qu'il atteigne l'arborescence souhaitée et permette d'écrire dans l'annuaire, à condition que vous n'ayez pas déjà renseigné le champ **Base DN** du composant **tLDAPOutput**.

4. Dans cet exemple, la variable globale **GetResultName** est utilisée pour récupérer automatiquement le chemin d'accès. Appuyez sur **Ctrl+Espace** pour accéder à la liste de variable et sélectionnez `tLDAPInput_1_RESULT_NAME`.

5. Dans le champ **Expression** de la colonne *mail*, saisissez l'adresse e-mail qui remplacera celle obsolète dans l'annuaire LDAP. Dans cet exemple, le nouvel e-mail est : *Pierre.Dupont@talend.com*.

   Cliquez sur **OK** pour valider les modifications.

6. Sélectionnez le composant **tLDAPOutput** pour définir les propriétés d'écriture dans l'annuaire.

7. Définissez manuellement les champs **Port** et **Host** si ces informations ne sont pas stockées dans le **Repository**.

8. Dans le champ **Base DN**, définissez la plus haute arborescence à laquelle vous avez accès. Si vous n'avez pas défini le chemin complet et exact au DN cible, définissez-le ici. Dans cet exemple, le DN complet est fournit par la sortie *dn* du composant **tMap**, Ainsi, seule la plus haute arborescence accessible est fournie : *o=directoryRoot*.

9. Sélectionnez le protocole à utiliser : **LDAP** pour cet exemple.

   Puis renseignez les champs **User** et **Password** attendus par l'annuaire LDAP.

   Dans le champ **Multi-valued field separator**, saisissez une virgule afin de séparer les champs contenant plusieurs valeurs séparées par des virgules.

10. Laissez les paramètres par défaut des champs **Alias Dereferencing** et **Referral Handling**, c'est-à-dire respectivement **Always** et **Ignore**.

    Dans le champ **Insert mode**, sélectionnez l'option **Update** pour cet exemple (l'adresse e-mail).

    Le schéma est récupéré du composant précédent à l'aide d'une opération de propagation.

11. Dans la vue **Advanced settings**, cochez la case **Use Attributes Options (for update mode)** pour afficher la table **Attribute Options**.

    Sélectionnez l'attribut **mail** dans la colonne **Attribute Name** et sélectionnez **Replace** dans la colonne **Option**.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Exécutez le Job en appuyant sur la touche **F6**.

```
Starting job LDAPInputnew at 14:17 20/09/2007.
uid=PIERRE DUPONT,ou=DATA,o=TALENDM,o=TALEND|PIERRE DUPONT|Pierre.Dupont@talend.com
Job LDAPInputnew ended at 14:17 20/09/2007. [exit code=0]
```

tLDAPRenameEntry

Propriétés du tLDAPRenameEntry

Famille de composant	Databases/LDAP	
Fonction	Le composant **tLDAPRenameEntry** renomme des entrées dans un annuaire LDAP.	
Objectif	Le **tLDAPRenameEntry** permet de renommer une ou des entrée(s) dans un annuaire LDAP sélectionné.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et cliquez sur le composant **tLDAPConnection** dans la liste **Component List**, afin de réutiliser les informations de la connexion que vous avez précédemment définie.
	Host	Adresse IP du serveur d'annuaire LDAP.
	Port	Numéro du port d'écoute du serveur.
	Base DN	Chemin d'accès à l'arborescence de l'utilisateur autorisé.
	Protocol	Sélectionnez le type de protocole dans la liste. **LDAP** : aucun codage n'est utilisé **LDAPS** : LDAP sécurisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît. Une fois sélectionné, le mode avancé vous permet de spécifier le répertoire et le mot de passe de la clé secrète du certificat pour le stockage d'un CA spécifique. Cependant, vous pouvez toujours désactiver la validation du certificat, en cochant la case **Trust all certs**. **TLS** : le certificat est utilisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît et est utilisée de la même façon que pour le type **LPDAPS**.
	User et *Password*	Renseignez les informations d'authentification de l'utilisateur. Notez que le login doit correspondre à la syntaxe LDAP pour être valide. Par exemple : "cn=Directory Manager". Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Alias dereferencing	Sélectionnez l'option dans la liste. **Never** permet l'amélioration des performances de recherche si vous êtes sûr qu'aucun alias n'est déréférencé. Par défaut, utilisez **Always** : **Always** : les alias sont toujours déréférencés. **Never** : les alias ne sont jamais déréférencés. **Searching** : cette option déréférence les alias uniquement après la résolution du nom.

		Finding : cette option déréférence les alias uniquement lors de la résolution du nom.
	Referrals handling	Sélectionnez l'option dans la liste : **Ignore** : ne tient pas compte des redirections des requêtes. **Follow** : tient compte des redirections des requêtes.
	Previous DN et *New DN*	Sélectionnez à partir de la liste déroulante la colonne du schéma portant l'ancien DN (**Previous DN**) et celle portant le nouveau DN à appliquer (**New DN**).
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**) , soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau des composants.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les requêtes LDAP possibles. Il est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d'autres composants. Note : Appuyez sur **Ctrl + Espace** pour accéder à la liste des variables globales, inclus la variable **GetResultName** permettant de récupérer automatiquement la base correspondante.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour un scénario associé au composant **tLDAPRenameEntry**, consultez :

- *Scénario : Afficher le contenu filtré d'un annuaire LDAP* du composant **tLDAPInput**.
- *Scénario : Editer des données dans un annuaire LDAP* du **tLDAPOutput**.

tMaxDBInput

Propriétés du tMaxDBInput

Famille de composant	Databases/MaxDB	
Fonction	Le composant **tMaxDBInput** lit une table de base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tMaxDBInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Host name	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table name	Saisissez le nom de la table.
	Query type et Query	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Guess schema	Cliquez sur le bouton pour récupérer le schéma de la table.
Advanced settings	*Trim all the String/Char columns*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez les scénarios dans :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

tMaxDBOutput

Propriétés du tMaxDBOutput

Famille de composant	Databases/MaxDB	
Fonction	Le composant **tMaxDBOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tMaxDBOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes.

		Insert or update : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tMaxDBOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données MaxDB. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tMaxDBRow

Propriétés du tMaxDBRow

Famille de composant	Databases/MaxDB	
Fonction	**tMaxDBRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tMaxDBRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur.
	Database	Nom de la base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

	Table name	Saisissez le nom de la table.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Guess Query	Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tParseRecordSet

Propriétés du tParseRecordSet

Ce composant se trouve à la racine de la famille **Databases** de la **Palette** de la perspective **Integration** du *Studio Talend*. Le composant **tParseRecordSet** couvre des besoins indirectement liés à l'utilisation des bases de données de tout type.

Famille de composant	Databases	
Fonction	Le composant **tParseRecordSet** analyse un ensemble d'enregistrements d'une table de données ou d'une requête SQL et retourne éventuellement des enregistrements seuls.	
Objectif	Il permet d'analyser un ensemble d'enregistrements d'une table au lieu de les analyser individuellement.	
Basic settings	*Prev. Comp. Column list*	Sélectionnez la colonne contenant l'ensemble d'enregistrements à analyser.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant uniquement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Attribute table	Paramétrez la valeur positionnelle de chaque colonne de chaque enregistrement d'un ensemble d'enregistrements.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d'entrée sont autorisés.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Ce composant est principalement utilisé avec la fonctionnalité Recordset d'un composant SP.

Scénario associé

Pour un exemple d'utilisation du **tParseRecordSet**, consultez *Scénario 2 : Utiliser l'instance PreparedStatement pour faire une requête sur des données*.

tPostgresPlusBulkExec

Propriétés du tPostgresPlusBulkExec

Les composants **tPostgresPlusOutputBulk** et **tPostgresPlusBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tPostgresPlusOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Famille de composant	Databases/PostgresPlus	
Fonction	Le composant **PostgresPlusBulkExec** exécute des opérations d'Insert sur les données fournies.	
Objectif	Le **tPostgresPlusBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données DB2.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification sur l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :

		None: n'effectuer aucune opération de table. **Drop and create table**: supprimer la table puis en créer une nouvelle. **Create table**: créer une table qui n'existe pas encore. **Create table if not exists**: créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table**: supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	File Name	Nom du fichier à charger. *Ce fichier est situé sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
Advanced settings	*Action*	Sélectionnez l'opération que vous voulez effectuer : **Bulk insert** ou **Bulk update**. Les informations demandées seront différentes en fonction de l'action choisie.
	Field terminated by	Caractère, chaîne ou expression régulière séparant les champs.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant.	

	La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes Postgresplus.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé au composant **tPostgresPlusBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tPostgresPlusClose

Propriétés du tPostgresPlusClose

Famille de composant	Databases/PostgresPlus	
Fonction	Le composant **tPostgresPlusClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données PostgresPlus.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tPostgresPlusConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tPostgresPlusConnection** et **tPostgresPlusRollback**. Il est généralement utilisé avec un composant **tPostgresPlusConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tPostgresPlusClose**, consultez *tMysqlConnection*.

tPostgresPlusCommit

Propriétés du tPostgresPlusCommit

Famille de composant	Databases/PostgresPlus	
Fonction	Le composant **tPostgresPlusCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tPostgresPlusConnection** dans la liste.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tPostgresPlusCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants PostgresPlus et notamment avec les composants **tPostgresPlusConnection** et **tPostgresPlusRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tPostgresPlusCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tPostgresPlusConnection

Propriétés du tPostgresPlusConnection

Famille de composant	Databases/PostgresPlus	
Fonction	Le composant **tPostgresPlusConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

		Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec des composants PostgresPlus et notamment avec les composants **tPostgresPlusCommit** et **tPostgresPlusRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tPostgresPlusConnection**, consultez *tMysqlConnection*.

tPostgresPlusInput

Propriétés du tPostgresPlusInput

Famille de composant	Databases/ PostgresPlus	
Fonction	Le composant **tPostgresPlusInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tPostgresPlusInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**) , soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.

		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Table name	Nom de la table de base de données.
	Query type et *Query*	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Use cursor*	Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d'optimiser les performances
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données PostgresPlus.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé, consultez les scénarios du composant **tPostgresPlusInput** :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

tPostgresPlusOutput

Propriétés du tPostgresPlusOutput

Famille de composant	Databases/ PostgresPlus	
Fonction	Le **tPostgresPlusOutput** écrit, met à jour, effectue les changements ou supprime les entrées dans une base de données.	
Objectif	Le **tPostgresPlusOutput** exécute l'action définie sur la table et/ou sur les données contenues dans la table, à partir du flux entrant du composant précédent dans le Job.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create table** : supprimer la table puis en créer une nouvelle.

		Create table : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in** , soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.

Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tPostgresqlOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	Support null in "SQL WHERE" statement	Cochez cette case pour prendre en compte les valeurs Null d'une table de base de données. Assurez-vous que la case **Nullable** est bien cochée pour les colonnes du schéma correspondantes.
	Use Batch size	Cochez cette case afin d'activer le mode batch pour le traitement des données. Dans le champ **Batch Size** qui apparaît, vous pouvez saisir le nombre de lignes par lot à charger. Cette case est disponible uniquement si vous avez choisi l'option **Insert**, **Update**, **Single Insert Query** ou **Delete** dans le champ **Action on data**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**.	

	Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données PostgresPlus. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tPostgresPlusOutputBulk

Propriétés du tPostgresPlusOutputBulk

Les composants **tPostgresPlusOutputBulk** et **tPostgresPlusBulkExec** sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant **tPostgresPlusOutputBulkExec**, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Famille de composant	Databases/PostgresPlus	
Fonction	Ecrit un fichier composé de colonnes et basé sur le délimiteur défini et sur les standards Postgresql.	
Objectif	Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Postgresql.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Nom du fichier à générer. *Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.*
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
Advanced settings	*Row separator*	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Include header	Cochez cette case pour inclure l'en-tête des colonnes dans le fichier.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	
Utilisation	Ce composant est généralement utilisé avec le composant **tPostgresPlusBulkExec**. Ensemble ils offrent un gain de performance important pour l'alimentation d'une base de données PostgresPlus.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé au composant **tPostgresPlusOutputBulk**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tPostgresPlusOutputBulkExec

Propriétés du tPostgresplusOutputBulkExec

Les composants **tPostgresPlusOutputBulk** et **tPostgresPlusBulkExec** sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant **tPostgresPlusOutputBulkExec**.

Famille de composant	Databases/Postgresplus	
Fonction	Le **tPostgresPlusOutputBulkExec** effectue une action d'Insert sur les données fournies.	
Objectif	Le **tPostgresPlusOutputBulkExec** est un composant dédié qui permet un gain de performance pendant les opérations d'Insert dans une base de données PostgresPlus.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de la base de données. Seuls *localhost*, *127.0.0.1* ou l'adresse IP exacte de la machine locale permettent un fonctionnement optimal. Le serveur de la base de données doit être installé sur la même machine que le *Studio Talend* ou le Job contenant un **tPostgresPlusOutputBulkExec**.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'Insert soit autorisée.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Clear a table** : supprimer le contenu de la table.
	File Name	Nom du fichier à générer et à charger.

		*Ce fichier est généré sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
Advanced settings	*Action*	Sélectionnez l'opération que vous voulez effectuer : **Bulk insert** ou **Bulk update**. Les informations demandées seront différentes en fonction de l'action choisie.
	File type	Sélectionnez le type de fichiers à traiter.
	Null string	Chaîne de caractères affichée pour signifier que la valeur est nulle.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Fields terminated by	Caractère, chaîne ou expression régulière séparant les champs : **None** : aucun caractère de séparation. **Whitespace** : le caractère de séparation est un espace. **EOF (used for loading LOBs from lobfile)** : la séparation est constituée d'un caractère de fin de fichier (End Of File). **Other terminator** : la séparation est constituée d'un caractère autre que ceux précédemment énoncés.
	Escape character	Caractère d'échappement de la ligne.
	Text enclosure	Caractère utilisé pour entourer le texte.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est principalement utilisé lorsqu'aucune transformation particulière n'est requise sur les données à charger dans la base de données.	

Scénarios associés

Pour un scénario associé au composant **tPostgresPlusOutputBulkExec**, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* du **tMysqlOutputBulk**.
- *Scénario : Insérer des données dans une base MySQL* du **tMysqlOutputBulkExec**.
- *Scénario : Supprimer et insérer des données dans une base Oracle* du **tOracleBulkExec**.

tPostgresPlusRollback

Propriétés du tPostgresPlusRollback

Famille de composant	Databases/PostgresPlus	
Fonction	Le **tPostgresPlusRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant permet d'éviter le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant **tPostgresPlusConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec d'autres composants PostgresPlus, notamment les composants **tPostgresPlusConnection** et **tPostgresPlusCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tPostgresPlusRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille*.

tPostgresPlusRow

Propriétés du tPostgresPlusRow

Famille de composant	Databases/ PostgresPlus	
Fonction	Le **tPostgresPlusRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, **tPostgresPlusRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Liste des versions de la base de données.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Database	Nom de la base de données.
	Schema	Nom exact du schéma
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Table name	Nom de la table de base à lire.
	Query type	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.
- *Scénario : Réinitialiser l'auto-incrémentation d'une base de données* du composant **tDBSQLRow**.
- *Scénario 1 : Supprimer et re-générer un index de table MySQL* du composant **tMysqlRow**.

tPostgresPlusSCD

Le composant **tPostgresPlusSCD** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tPostgresPlusSCD*.

tPostgresPlusSCDELT

Le composant **tPostgresPlusSCDELT** appartient à deux familles différentes : Business Intelligence et Databases. Pour plus d'informations, consultez *tPostgresPlusSCDELT*.

tSAPHanaClose

Propriétés du tSAPHanaClose

Famille de composant	Databases / SAP Hana	
Fonction	Le composant **tSAPHanaClose** ferme la connexion ouverte à la base de données après fin de la transaction.	
Objectif	Le composant ferme une connexion à la base de données SAP Hana.	
Basic settings	*Component list*	Sélectionnez le composant **tSAPHanaConnection** dans la liste si plus d'une connexion est définie dans le Job.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour connecter les données de log au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec d'autres composants SAP Hana, notamment le **tSAPHanaConnection** et le **tSAPHanaCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	n/a	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSAPHanaCommit

Propriétés du tSAPHanaCommit

Famille de composant	Databases / SAP Hana	
Fonction	Le composant **tSAPHanaCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	Sélectionnez le composant **tSAPHanaConnection** dans la liste si plus d'une connexion est définie dans le Job.
	Close Connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tSAPHanaCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour connecter les données de log au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec d'autres composants SAP Hana, notamment avec le **tSAPHanaConnection** et le **tSAPHanaRollback**. Utilisez ce composant si l'option **Auto Commit** du **tSAPHanaConnection** est décochée.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	n/a	

Scénario associé

Pour un scénario associé à ce composant, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tSAPHanaConnection

Propriétés du tSAPHanaConnection

Famille de composant	Databases / SAP Hana	
Fonction	Le composant **tSAPHanaConnection** ouvre une connexion à la base de données SAP Hana afin d'effectuer une transaction.	
Objectif	Ce composant vous permet d'établir une connexion à SAP Hana à réutiliser dans d'autres composants SAP Hana au sein du Job.	
Basic settings	*DB Version*	Sélectionnez la version de la base de données SAP Hana Database (HDB) que vous utilisez.
	Host	Saisissez l'adresse IP du serveur de la base de données.
	Port	Saisissez le numéro du port d'écoute du serveur de la base de données.
	Table Schema	Saisissez le nom du schéma de la base de données.
	Username et *Password*	Saisissez vos informations d'authentification à la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Additional JDBC Parameters	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log lors du traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Utilisation	Ce composant est généralement utilisé avec d'autres composants SAP Hana, notamment le **tSAPHanaCommit** et le **tSAPHanaRollback**, par exemple pour fermer une connexion à la fin d'une transaction.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	n/a	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSAPHanaInput

Propriétés du tSAPHanaInput

Famille de composant	Databases / SAP Hana	
Fonction	Le composant **tSAPHanaInput** lit une base de données et extrait des champs selon une requête.	
Objectif	Le **tSAPHanaInput**exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérés est ensuite transmise au composant suivant via une connexion de flux **Main** > **Row**.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version de SAP Hana Database (HDB) que vous utilisez.
	Host	Saisissez l'adresse IP du serveur de la base de données.
	Port	Saisissez le numéro du port d'écoute du serveur de la base de données.
	Schema	Nom du schéma de la base de données.
	Username et *Password*	Saisissez vos informations d'authentification à la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

	Table Name	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Query Type	Peut être **Built-in** ou **Repository**.. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement l'instruction de requête ou construisez-la graphiquement via le SQLBuilder.
		Repository : Sélectionnez la requête stockée dans le **Repository**. Le champ **Query** est automatiquement renseigné.
	Guess Query	Cliquez sur le bouton **Guess Query** afin de générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Guess schema	Cliquez sur le bouton **Guess schema** pour récupérer le schéma de la table.
	Query	Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.
Advanced settings	*Additional JDBC Parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Cette option est indisponible lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**.
	Trim all the String/Char columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes de type String/Char.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées. Décochez la case **Trim all the String/Char columns** pour activer cette option.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en standalone car il comprend le moteur de SAP Hana. C'est un composant de début pouvant initier un traitement de flux de données.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSAPHanaOutput

Propriétés du tSAPHanaOutput

Famille de composant	Databases / SAP Hana	
Fonction	Le composant **tSAPHanaOutput** écrit, met à jour, modifie ou supprime des entrées dans une base de données SAP Hana.	
Objectif	Le **tSAPHanaOutput** exécute l'action définie sur la table et/ou sur les données contenues dans la table, à partir du flux entrant du composant précédent dans le Job.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version de SAP Hana Database (HDB) que vous utilisez.
	Host	Saisissez l'adresse IP du serveur de la base de données.
	Port	Saisissez le numéro du port d'écoute du serveur de la base de données.
	Table Schema	Saisissez le nom du schéma de la base de données.
	Username et *Password*	Saisissez vos informations d'authentification à la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Saisissez le nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Action on table	Sur la table définie, vous pouvez effectuer l'une des opérations suivantes : **None** : Aucune opération n'est effectuée. **Drop and create table** : La table est supprimée et créée à nouveau. **Create table** : La table n'existe pas et est créée. **Create table if not exists** : La table est créée si elle n'existe pas. **Drop table if exists and create** : La table est supprimée si elle existe déjà puis créée à nouveau. **Clear table** : Le contenu de la table est supprimé.
	Type of table storage	Cette option n'est disponible que si vous créez (en supprimant ou pas) la table de la base de données. Cette option vous permet de définir la manière dont les données sont stockées dans la table. Les types d'organisation du stockage des données suivants sont disponibles : • **Row** : Les données sont stockées sous forme de lignes. Il est préférable d'utiliser ce type de table si la majorité des accès à la table consiste en la sélection de quelques enregistrements avec tous les attributs sélectionnés.

		• **Column** : Les données sont stockées sous forme de colonnes. Il est préférable d'utiliser ce type de table si la majorité des accès à la table s'effectue un grand nombre de fois avec seulement quelques attributs sélectionnés. • **History column** : Une table avec un type de session de transaction particulier, *HISTORY* est créée. Les tables avec le type de session *HISTORY* supportent les requêtes de type *time travel*. Pour plus d'informations concernant *HISTORY* et les requêtes *time travel*, consultez http://help.sap.com/saphelp_hanaplatform/helpdata/en/20/d56cd075191014b7ea9e4e77477aae/content.htm (en anglais). • **Global temporary** : La définition de la table et les données peuvent être vues par tous les utilisateurs et toutes les connexions pendant la durée de la session en cours. La table est supprimée à la fin de la session. • **Local temporary** : La définition de la table et les données ne peuvent être vues que par l'utilisateur qui a créé la table pendant la durée de la session en cours. La table est supprimée à la fin de la session.
	Action on data	Sur les données de la table définie, vous pouvez choisir d'effectuer l'opération suivante : **Insert** : Ajouter de nouvelles entrées à la table. Si des doublons sont trouvés, le Job s'arrête. **Update** : Apporter des modifications à des entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer des entrées correspondant au flux d'entrée. *Il est nécessaire de spécifier au moins une colonne en tant que clé primaire sur laquelle baser les opérations **Update** et **Delete**. Vous pouvez faire ceci en cliquant sur **Edit Schema** et en cochant la case correspondant à la colonne (aux colonnes) que vous souhaitez définir comme clé primaire. Pour une utilisation avancée, consultez la description de l'option **Use field options**, dans l'onglet **Advanced settings** de ce composant.*
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

	Die on error	Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur. Si nécessaire, vous pouvez récupérer les lignes en erreur via un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC Parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Cette option est indisponible lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**.
	Commit every	Saisissez le nombre de lignes à terminer avant de commiter des lots de lignes en même temps dans la base de données. Cette option vous assure une certaine qualité de transaction (mais pas de rollback) et, surtout, une meilleure performance à l'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tSAPHanaOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données. Dans cette table, vous pouvez : **Key in update** : définir la clé primaire à utiliser lors d'une opération de mise à jour. **Key in delete** : définir la clé primaire à utiliser lors d'une opération de suppression. **Updatable** : définir les colonnes à mettre à jour. **Insertable** : définir les colonnes à insérer.
	Enable debug mode	Cochez cette case pour afficher chaque étape durant le traitement des entrées dans une base de données.
	Support null in "SQL WHERE" statement	Cochez cette case pour prendre en compte les valeurs nulls dans une instruction "SQL WHERE".
	Use Batch Size	Lorsque cette case est cochée, saisissez le nombre de lignes contenues dans chaque lot traité. *Cette option est disponible uniquement lorsque la case **Use an existing connection** n'est pas cochée dans la vue **Basic settings**.*
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier.	

	QUERY : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant doit être relié à un composant d'entrée. Il vous permet d'effectuer des actions sur une table ou sur les données d'une de la base de données SAP Hana. Cela vous permet de créer des flux de rejet à l'aide d'un lien **Row > Reject** afin de filtrer les données en erreur.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSAPHanaRollback

Propriétés du tSAPHanaRollback

Famille de composant	Databases / SAP Hana	
Fonction	Le composant **tSAPHanaRollback** annule une transaction dans la base de données connectée.	
Objectif	Ce composant permet d'éviter le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez dans la liste le composant **tSAPHanaConnection** si plus d'une connexion est définie dans le Job.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion sélectionnée une fois que le composant a terminé ses tâches.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est généralement utilisé avec d'autres composants SAP Hana, notamment le **tSAPHanaConnection** et **tSAPHanaCommit**, afin de mener à bien des transactions.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	n/a	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSAPHanaRow

Propriétés du tSAPHanaRow

Famille de composant	Databases / SAP Hana	
Fonction	Le composant **tSAPHanaRow** est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.	
Objectif	Selon la nature de la requête et de la base de données, le **tSAPHanaRow** agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	DB Version	Sélectionnez la version de SAP Hana Database (HDB) que vous utilisez.
	Property type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro du port d'écoute du serveur de base de données.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.
	Query Type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Guess Query	Cliquez sur le bouton **Guess Query** afin de générer la requête correspondant au schéma de votre table dans le champ **Query**.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Additional JDBC Parameters*	Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Cette option est indisponible si vous avez coché la case **Use an existing connection** dans la vue **Basic settings**.
	Propagate QUERY's recordset	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Commit every	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSasInput

*Assurez-vous d'installer au préalable les modules sas.core.jar, sas.intrnet.javatools.jar et sas.svc.connection.jar sous le chemin **lib** > **java** dans votre build du Studio Talend pour pouvoir utiliser toutes les fonctionnalités des composants SAS. Vous pouvez vérifier que ces modules ont bien été installés à partir de la vue **Modules** de votre Studio Talend.*

Propriétés du tSasInput

Famille de composant	Databases/SAS	
Fonction	Le composant **tSasInput** lit une base de données et en extrait des champs à l'aide de requêtes.	
Objectif	Le **tSasInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Host name	Nom ou adresse IP du serveur SAS.
	Port	Port d'écoute à ouvrir.
	Librefs	Renseignez le nom du répertoire physique dans lequel est contenue la table à lire, suivi de son chemin d'accès. Par exemple : *"TpSas 'C:/SAS/TpSas'"*
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Renseignez le nom de la table à lire, préfixé par le nom du répertoire physique dans lequel elle est contenue. Par exemple : `"TpSas.Customers"`.
	Query type	La requête peut être **Built-in** ou distante dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
	Query	Saisissez votre requête, si elle n'est pas stockée dans le **Repository**, en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données utilisant une connexion SAS.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez les scénarios du composant **tDBInput** :

- *Scénario 1 : Afficher les données sélectionnées à partir d'une table*.
- *Scénario 2 : Utiliser la variable StoreSQLQuery*.

Consultez également *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement* du composant **tContextLoad**.

tSasOutput

*Assurez-vous d'installer au préalable les modules sas.core.jar, sas.intrnet.javatools.jar et sas.svc.connection.jar sous le chemin **lib** > **java** dans votre build du Studio Talend pour pouvoir utiliser toutes les fonctionnalités des composants SAS. Vous pouvez vérifier que ces modules ont bien été installés à partir de la vue **Modules** de votre Studio Talend.*

Propriétés du tSasOutput

Famille de composant	Databases/SAS	
Fonction	Le composant **tSasOutput** écrit, met à jour, modifie ou supprime les données d'une base de données.	
Objectif	Le **tSasOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	SAS URL	Entrez le chemin d'accès à la base de données.
	Driver JAR	Sélectionnez le driver JAR à partir de la liste déroulante ou cliquez sur les trois points **[...]** pour ajouter un nouveau JAR à la liste.
	Class Name	Dans le champ **Class name**, renseignez le nom de la classe pour qu'elle soit indiquée dans le driver.
	Username et *Password*	Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Table	Nom de la table à lire.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton*

		[...] *à côté du champ* ***Edit Schema*** *et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet* ***Advanced settings*** *pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case* ***Use field options*** *et sélectionnez la case* ***Key in update*** *correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases* ***Key in delete*** *pour les opérations de suppression (Delete).*
	Clear data in table	Cochez cette case pour que les données soient supprimées de la table sélectionnée avant toute action.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.
		Reference column : Saisissez une colonne de référence que le composant **tSasOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.

	Enable debug mode	Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant couvre toutes les possibilités de requête SQL dans les bases de données utilisant une connexion SAS. Il permet de faire des actions sur une table ou les données d'une table d'une base de données SAS. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé au **tSasOutput**, consultez :

- *Scénario : Ecrire une ligne dans une table d'une base de données MySQL via une connexion ODBC* du composant **tDBOutput**.
- *Scénario 1 : Ajouter une colonne et modifier les données* du composant **tMysqlOutput**.

tSQLiteClose

Propriétés du tSQLiteClose

Famille de composant	Databases/SQLite	
Fonction	Le composant **tSQLiteClose** ferme la connexion à une base de données connectée.	
Objectif	Ce composant permet de fermer une connexion à la base de données SQLite.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tSQLiteConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tSQLiteConnection** et **tSQLiteRollback**. Il est généralement utilisé avec un composant **tSQLiteConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tSQLiteClose**, consultez *tMysqlConnection*.

tSQLiteCommit

Propriétés du tSQLiteCommit

Famille de composant	Databases/SQLite	
Fonction	Le composant **tSQLiteCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component list*	Sélectionnez le composant **tSQLiteConnection** dans la liste s'il y a plus d'une connexion dans votre Job.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tSQLiteCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants SQLite, notamment les composants **tSQLiteConnection** et **tSQLiteRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tSQLiteCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*.

tSQLiteConnection

Propriétés du tSQLiteConnection

Famille de composant	Databases/SQLite	
Fonction	Le composant **tSQLiteConnection** ouvre une connexion vers une base de données afin d'effectuer une transaction.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Database	Nom de la base de données.
	Table Schema	Nom du schéma.
	Use or register a shared DB Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.* **Shared DB Connection Name :** définissez ou saisissez le nom de la connexion partagée.
Advanced settings	*Auto Commit*	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.

Utilisation	Ce composant est généralement utilisé avec des composants SQLite, notamment les composants **tSQLiteCommit** et **tSQLiteRollback**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour un scénario associé au composant **tSQLiteConnection**, consultez *tMysqlConnection*.

tSQLiteInput

Propriétés du tSQLiteInput

Famille de composant	Databases/ SQLite	
Fonction	Le composant **tSQLiteInput** lit une base de données et en extrait des champs à l'aide de requêtes. Un petit moteur de base de données SQLite lui est intégré, il n'a donc pas besoin d'être connecté à un serveur de base de données.	
Objectif	**Le tSQLiteInput** exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Database	Chemin d'accès au fichier de la base de données SQLite.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Query type	La requête peut être construite manuellement (Built-in) pour un Job particulier ou stockée dans le Repository s'il s'agit d'une requête fréquemment utilisée, pour en faciliter l'utilisation.
	Query	Si elle n'est pas stockée dans le **Repository**, saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
Advanced settings	*Trim all the String/Char columns*	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	Trim column	Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	tStat Catcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est tout à fait autonome du fait de son moteur de base de données SQLite intégré. Il s'agit d'un composant de début pouvant initier un traitement de flux de données.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Filtrer des données SQlite

Le scénario suivant décrit un Job assez simple qui permet d'appliquer une requête donnée utilisant un filtre pour extraire des lignes à partir d'une base de données SQLite source et alimenter une table SQLite en sortie.

- Cliquez-déposez à partir de la **Palette** les composants **tSQLiteInput** et **tSQLiteOutput**.
- Reliez le composant d'entrée et de sortie à l'aide d'un lien de type **row Main**.
- Dans l'onglet **Basic settings** du composant **tSQLiteInput**, saisissez le chemin d'accès ou parcourez vos dossiers jusqu'au fichier d'entrée à la base de donnée SQLite.

	id	version	download_date	ip	type	type_os
1	20027	TOS-Win32-200610	13/11/2006	1191947907	1	101
2	20028	TOS-Win32-200610	13/11/2006	1195650472	1	102
3	20030	TOS-Win32-200610	13/11/2006	3375565745	1	103
4	20031	TOS-Win32-200610	13/11/2006	1195650472	1	104
5	20032	TOS-Win32-200610	13/11/2006	1195650472	1	105
6	20033	TOS-Win32-200610	13/11/2006	1104872453	1	106
7	20034	TOS-Win32-200610	13/11/2006	1104872453	1	107
8	20036	TOS-Win32-200610	13/11/2006	1190898057	1	108
9	20037	TOS-Win32-200610	13/11/2006	1190898057	1	109
10	20038	TOS-Win32-200610	13/11/2006	1348977142	1	110
11	20040	TOS-Win32-200610	13/11/2006	3581349521	1	11
12	20041	TOS-Win32-200610	13/11/2006	1190898057	1	12
13	20043	TOS-Win32-200610	13/11/2006	1196485544	1	13
14	20044	TOS-Win32-200610	13/11/2006	1066743463	1	14
15	20045	TOS-Win32-200610	13/11/2006	1196485544	1	15
16	20046	TOS-Win32-200610	13/11/2006	2624217794	1	16

- Le fichier contient des centaines de lignes et comprend une colonne **ip** servant de base à la requête.
- Dans l'onglet **Basic settings** du composant **tSQLiteInput**, modifiez le schéma afin qu'il corresponde à la structure de la table.

- Dans le champ **Query**, renseignez la requête choisie par rapport à la colonne *ip*.
- Sélectionnez les paramètres d'encodage correspondants.
- Pour le composant **tSQLiteOutput**, dans le champ **Database** de l'onglet **Basic settings**, sélectionnez le chemin d'accès à la base de données de sortie.

- Dans le champ **Table**, renseignez le nom de la table à alimenter avec les données sélectionnées.
- A partir des listes déroulantes **Action on table** et **Action on Data**, sélectionnez les actions à effectuer sur la table et sur les données. Dans le cas présent, les actions sont, respectivement, *Drop and create* et *Insert*.
- Synchronisez le schéma avec le schéma d'entrée.
- Sélectionnez l'encodage puis définissez la limite de commit.
- Enregistrez le Job puis exécutez-le en appuyant sur **F6**.

	id	version	download_date	ip	type	type_os
1	20028	TOS-Win32-200610	13/11/2006	1195650472	1	102
2	20031	TOS-Win32-200610	13/11/2006	1195650472	1	104
3	20032	TOS-Win32-200610	13/11/2006	1195650472	1	105

Les données sélectionnées sont renvoyées vers le fichier SQLite défini.

tSQLiteOutput

Propriétés du tSQLiteOutput

Famille de composant	Databases/ SQLite	
Fonction	Le composant **tSQLiteOutput** écrit, met à jour, modifie ou supprime les données d'une base de données. Un petit moteur de base de données SQLite lui est intégré, il n'a donc pas besoin d'être connecté à un serveur de base de données.	
Objectif	Le **tSQLiteOutput** exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
		Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	Database	Chemin d'accès au fichier de la base de données.
	Table	Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.
	Action on table	Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : **None** : n'effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n'existe pas encore. **Create table if doesn't exist** : créer la table si nécessaire. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.
	Action on data	Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

		Insert : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons. **Update** : Mettre à jour les entrées existantes. **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré. **Delete** : Supprimer les entrées correspondantes au flux d'entrée. *Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).*
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut, qui sont des entiers (integer) ou des fonctions (function), assurez-vous que les valeurs par défaut ne sont pas entourées de guillemets. Si c'est le cas, vous devez supprimer manuellement les guillemets. Pour plus d'informations, consultez https://help.talend.com/display/KB/Verifying+default+values+in+a+retrieved+schema (en anglais).
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Commit every*	Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.
	Additional Columns	Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne à modifier ou à insérer.
		SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
		Position : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.

		Reference column : Saisissez une colonne de référence que le composant **tSQLiteOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
	Use field options	Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.
	Enable debug mode	Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit absolument être relié à un composant d'entrée (Input). Il permet de faire des actions sur une table ou les données d'une table d'une base de données SQLite. Il permet aussi de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données en erreur. Pour un exemple d'utilisation, consultez *Scénario 3 : Récupérer les données erronées à l'aide d'un lien Reject* du composant **tMysqlOutput**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au composant **tSQLiteOutput**, consultez *tSQLiteInput*.

tSQLiteRollback

Propriétés du tSQLiteRollback

Famille de composant	Databases/SQLite	
Fonction	Le **tSQLiteRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component list*	Sélectionnez le composant de connexion **tSQLiteConnection** dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en association avec des composants SQLite, notamment avec le **tSQLiteConnection** et le **tSQLiteCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tSQLiteRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMysqlRollback**.

tSQLiteRow

Propriétés du tSQLiteRow

Famille de composant	Databases/SQLite	
Fonction	Le **tSQLiteRow** exécute une requête définie dans une base de données spécifiée et utilise les paramètres de la colonne.	
Objectif	Une requête prédéfinie utilise le flux d'entrée pour remplacer les paramètres substituables par les valeurs définies pour chaque paramètre. Ce composant est très utile quand il s'agit de faire des mises à jour.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	Database	Chemin d'accès au fichier de la base de données SQLite.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
		Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant,

		sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Table Name	Nom de la table à lire.
	Query type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
		Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.
	Query	Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Propagate QUERY's recordset*	Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du **tParseRecordSet**.
	Use PreparedStatement	Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**. **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL. **Parameter Type** : Saisissez le type du paramètre. **Parameter Value** : Saisissez la valeur du paramètre. Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
	Commit every	Nombre de lignes avant le commit.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Mettre à jour des lignes SQLite

Le scénario suivant décrit un Job qui permet de mettre à jour un fichier de base de donnée SQLite en utilisant une requête prédéfinie et un fichier délimité.

- Cliquez-déposez les composants **tFileInputDelimited** et **tSQLiteRow**.
- Dans l'onglet **Basic settings** du composant **tFileInputDelimited**, parcourez les dossiers et sélectionnez le fichier d'entrée à utiliser pour mettre à jour les lignes dans la base de données.

- Il n'y a ni en-tête (**Header**) ni pied de page (**Footer**). Le séparateur de lignes (**Row separator**) est un retour chariot et le séparateur de champs (**Field separator**) un point-virgule.
- Modifiez le schéma s'il n'est pas stocké dans le **Repository**.

- Assurez-vous que la longueur et le type définissent correctement les colonnes.

Dans l'espace de modélisation graphique, double-cliquez sur le composant **tSQLiteRow** pour afficher sa vue **Component**.

- Dans l'onglet **Basic settings** du composant **tSQLiteRow**, renseignez le champ **Database** avec le chemin d'accès au fichier à mettre à jour.

- Le schéma est en mode read-only puisqu'il doit correspondre au schéma d'entrée.
- Saisissez la requête ou récupérez-la à partir du **Repository**. Dans le cas présent, la colonne *type_os* a été mise à jour en fonction de la valeur *id* définie dans le flux entrant. La requête se présente comme suit :

```
"Update download set type_os=? where id=?"
```

- Dans l'onglet **Advanced settings**, cochez la case **Use PreparedStatement** pour afficher la table des paramètres substituables.

- Dans la table des paramètres d'entrée, ajoutez autant de lignes qu'il y a de paramètres substituables. Dans cet exemple, définissez *type_os* et *id*.
- Renseignez le champ **Commit every**.
- Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

La table *download* de la base de données SQLite est ainsi mise à jour avec le nouveau code *type_os*, selon le fichier délimité d'entrée.

Scénario associé

Pour un scénario associé, consultez :

- *Scénario 3 : Combiner deux flux pour une sortie sélective*.

Composants DotNET

Ce chapitre contient les composants que vous pouvez trouver dans la famille **DotNET** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille DotNET comprend les connecteurs les plus fréquemment utilisés pour communiquer avec des objets .NET.

tDotNETInstantiate

Propriétés du tDotNETInstantiate

Famille de composant	DotNET	
Fonction	Le composant **tDotNETInstantiate** instancie un objet dans .NET pour une utilisation ultérieure.	
Objectif	Le **tDotNETInstantiate** invoque le constructeur d'un objet .NET pour un usage ultérieur.	
Basic settings	*Dll to load*	Saisissez dans ce champ le chemin d'accès à une bibliothèque DLL contenant la (les) classe(s) qui vous intéresse(nt) ou saisissez le nom de l'Assembly ou cliquez sur le bouton **[...]** afin de parcourir votre système jusqu'au répertoire de votre bibliothèque. Par exemple, *System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089* pour une Assembly OleDb.
	Fully qualified class name(i.e. ClassLibrary1.NameSpace2.Class1)	Renseignez ce champ en saisissant le chemin complet de la classe (FQN).
	Value(s) to pass to the constructor	Cliquez sur le bouton **[+]** pour ajouter une ou plusieurs valeur(s) à passer au constructeur de l'objet. Ou, laissez vide ce tableau afin d'appeler un constructeur par défaut pour l'objet. La/les valeur(s) valide(s) doi(ven)t être les paramètres requis par la classe à utiliser.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**INSTANCE** : instance d'un objet .NET. Cette variable est une variable **After** et retourne un object. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé en tant que composant de début dans un flux ou dans un sous-job indépendant. Pour utiliser ce composant, vous devez d'abord installer les fichiers DLLs d'exécution, par exemple *janet-win32.dll*, sous Windows 32 bits et *janet-win64.dll* sous Windows 64 bits, dans le package redistribuable Microsoft Visual C++, afin d'éviter les erreurs telles que *UnsatisfiedLinkError* sur les DLL dépendants. Assurez-vous que le DLL d'exécution et tous les autres DLL dont dépend le DLL à appeler sont entièrement installés et que leurs versions sont en adéquation. Les DLL d'exécution requis peuvent être installés dans le dossier *System32* ou dans le dossier *bin* du JRE à utiliser. Si vous devez exporter un Job utilisant ce composant pour l'exécuter hors du *Studio Talend*, vous devez spécifier le conteneur d'exécution souhaité en configurant l'argument *-Djava.library.path*.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Utiliser .NET dans Talend* du composant **tDotNETRow**.

tDotNETRow

Propriétés du tDotNETRow

Famille de composant	DotNET	
Fonction	Le composant **tDotNETRow** envoie des données à partir de bibliothèques et vers des bibliothèques et des classes dans .NET ou d'autres fichiers DLL personnalisés.	
Objectif	Le **tDotNETRow** vous permet de faciliter la transformation de données à l'aide de classes .NET built-in ou personnalisées.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant (**Repository)**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Use a static method	Cochez cette case pour invoquer une méthode statique dans .NET. Cela désactivera la case **Use an existing instance**.
	Propagate a data to output	Cochez cette case afin de propager des données transformées vers la sortie
	Use an existing instance	Cochez cette case pour réutiliser une instance existante d'un objet .NET dans la liste **Existing instance to use**. **Existing instance to use** : Sélectionnez une instance existante d'objets .NET créés par d'autres composants .NET de la liste. Cette case est désactivée si vous avez coché **Use a static method**. Cochez cette case désactivera les champs **Dll to load**, **Fully qualified class name(i.e. ClassLibrary1.NameSpace2.Class1)** et **Value(s) to pass to the constructor**.
	Dll to load	Saisissez dans ce champ le chemin d'accès à une bibliothèque DLL contenant la (les) classe(s) qui vous intéresse(nt) ou saisissez le nom de l'Assembly ou cliquez sur le bouton **[...]** afin de parcourir votre système jusqu'au répertoire de votre bibliothèque. Par exemple, *System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089* pour une Assembly OleDb.
	Fully qualified class name(i.e. ClassLibrary1.NameSpace2.Class1)	Renseignez ce champ en saisissant le chemin complet de la classe (FQN).
	Method name	Renseignez ce champ en saisissant le nom de la méthode à invoquer dans .NET.
	Value(s) to pass to the constructor	Cliquez sur le bouton **[+]** pour ajouter une ou plusieurs valeur(s) à passer au constructeur de l'objet. Ou, laissez vide ce tableau afin d'appeler un constructeur par défaut pour l'objet. La/les valeur(s) valide(s) doi(ven)t être les paramètres requis par la classe à utiliser.
	Method Parameters	Cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) de paramètres à passer à la méthode.

	Output value target column	Dans la liste, sélectionnez une colonne de la lignes de sortie afin de lui attribuer une valeur.
Advanced settings	*Create a new instance at each row*	Cochez cette case pour créer une nouvelle instance à chaque ligne qui passe dans le composant.
	Method doesn't return a value	Cochez cette case afin d'invoquer une méthode sans retourner de valeur comme résultat du traitement.
	Returns an instance of a .NET Object	Cochez cette case pour retourner une instance d'un objet .NET comme résultat d'une méthode invoquée.
	Store the returned value for later use	Cochez cette case afin de stocker la valeur retournée d'une méthode pour une utilisation ultérieure dans un autre composant **tDotNETRow**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé pour communiquer avec des objets .NET. Pour utiliser ce composant, vous devez d'abord installer les fichiers DLLs d'exécution, par exemple *janet-win32.dll*, sous Windows 32 bits et *janet-win64.dll* sous Windows 64 bits, dans le package redistribuable Microsoft Visual C++, afin d'éviter les erreurs telles que *UnsatisfiedLinkError* sur les DLL dépendants. Assurez-vous que le DLL d'exécution et tous les autres DLL dont dépend le DLL à appeler sont entièrement installés et que leurs versions sont en adéquation. Les DLLs d'exécution requis peuvent être installés dans le dossier *System32* ou dans le dossier *bin* du JRE à utiliser. Si vous devez exporter un Job utilisant ce composant pour l'exécuter hors du *Studio Talend*, vous devez spécifier le conteneur d'exécution souhaité en configurant l'argument *-Djava.library.path*.	

Scénario : Utiliser .NET dans Talend

Ce scénario décrit un Job à trois composants utilisant une bibliothèque DLL contenant une classe appelée *Test1.Class1*. On invoque une méthode de cette classe qui traite la valeur et écrit les données dans la console.

Prérequis

Avant de reproduire ce scénario, vous devez d'abord construire votre environnement d'exécution.

- Créez le DLL à charger par le **tDotNETInstantiate**

  Cette classe d'exemple construite dans .NET se présente comme suit :

```
using System;
using System.Collections.Generic;
using System.Text;


namespace Test1
{
    public class Class1
    {
        string s = null;
        public Class1(string s)
        {
            this.s = s;
        }

        public string getValue()
        {
            return "Return Value from Class1: " + s;
        }

    }
    }
```

  Cette classe lit la valeur d'entrée et ajoute le texte *Return Value from Class1:* devant cette valeur. La classe est compilée à l'aide du dernier .NET.

- Installez le fichier DLL d'exécution à partir du dernier .NET. Dans ce scénario, utilisez *janet-win32.dll* sous Windows 32 bits et placez-le dans le dossier *System32*.

  Le DLL d'exécution est compatible avec le DLL à charger.

Relier les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tDotNETInstantiate**, **tDotNETRow** et **tLogRow**.

2. Connectez le **tDotNETInstantiate** au **tDotNETRow** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

3. Reliez le **tDotNETRow** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer le tDotNETInstantiate

1. Double-cliquez sur le **tDotNETInstantiate** pour afficher sa vue **Basic settings** et définir ses propriétés de base.

2. Cliquez sur le bouton **[...]** à côté du champ **Dll to load** et parcourez votre système jusqu'au fichier dll à charger. Vous pouvez également renseigner le champ avec une Assembly. Dans cet exemple, utilisez :

   "C:/Program Files/ClassLibrary1/bin/Debug/ClassLibrary1.dll""

3. Dans le champ **Fully qualified class name**, saisissez un nom de classe valide à utiliser. Dans cet exemple, saisissez :

   "Test1.Class1"

4. Cliquez sur le bouton **[+]** sous la table **Value(s) to pass to the constructor** afin d'ajouter une ligne pour la valeur à passer au constructeur.

   Dans cet exemple, saisissez :

   `"Hello world"`

Configurer le composant tDotNETRow

1. Double-cliquez sur le composant **tDotNETRow** afin d'afficher sa vue **Basic settings** et définir ses propriétés de base.

2. Cochez la case **Propagate data to output**.

3. Cochez la case **Use an existing instance** et sélectionnez **tDotNETInstantiate_1** dans la liste **Existing instance to use**.

4. Renseignez le champ **Method Name** avec le nom de la méthode à utiliser. Dans cet exemple, utilisez *"getValue"*.

5. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ajouter une colonne au schéma.

Cliquez sur le bouton **[+]** afin d'ajouter une colonne au schéma et cliquez sur **OK** pour sauvegarder les modifications.

6. Sélectionnez **newColumn** dans la liste **Output value target column**.

7. Laissez les autres paramètres tels qu'ils sont.

Configurer le tLogRow

1. Double-cliquez sur le **tLogRow** afin d'afficher sa vue **Basic settings** et définir ses propriétés de base.

2. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma défini dans le composant précédent.

3. Sélectionnez **Table** dans la zone **Mode**.

Sauvegardez votre Job et appuyez sur la touche **F6** pour l'exécuter.

Vous pouvez constater, dans les résultats, que le texte `Return Value from Class1` est ajouté devant la valeur récupérée `Hello world`.

Composants ELT

Ce chapitre passe en revue les propriétés des principaux composants présents dans la famille **ELT** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille ELT regroupe les connecteurs de bases de données les plus courants, ainsi que des composants de traitement, tous dédiés au mode ELT dans lequel les SGBD cibles deviennent des outils de transformation.

Ce mode supporte les bases de données les plus courantes, comme Teradata, Oracle, Vertica, Netezza, Sybase, etc.

tAccessConnection

Le composant **AccessConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tAccessConnection*.

tAS400Connection

Le composant **tAS400Connection** appartient à deux familles différentes différentes : Databases et ELT. Pour plus d'informations, consultez *tAS400Connection*.

tCombinedSQLAggregate

Propriétés du tCombinedSQLAggregate

Famille de composant	ELT/CombinedSQL	
Fonction	Le composant **tCombinedSQLAggregate** collecte les valeurs de données d'une ou plusieurs colonnes d'une table dans des buts statistiques. Ce composant exécute en temps réel des transformations de données dans le SGBD lui-même.	
Objectif	Ce composant fournit un ensemble de matrices basées sur des valeurs ou sur des calculs.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Group by	Définit les ensembles d'agrégation, dont les valeurs sont utilisées pour les calculs.
		Output Column : Sélectionnez le libellé de colonne dans la liste fournie, basée sur la structure de schéma que vous avez définie. Vous pouvez ajouter autant de colonnes de sortie que vous le souhaitez afin d'affiner les agrégations.
		Input Column : Sélectionnez les libellés des colonnes d'entrée afin de les faire correspondre au contenu attendu des colonnes de sortie, dans le cas où vous souhaitez que les libellés du schéma de sortie soient différents du schéma d'entrée.
	Operations	Sélectionnez le type d'opération et la valeur à utiliser pour le calcul et le champ de sortie.
		Output Column : Sélectionnez le champ de destination dans la liste.
		Function : Sélectionnez l'une des opérations suivantes à effectuer sur les données : **count**, **min**, **max**, **avg**, **sum**, **first**, **last**, **distinct** et **count (distinct)**.

		Input column : Sélectionnez la colonne d'entrée à partir de laquelle vous souhaitez collecter les valeurs à agréger.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **Flow** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé en tant que composant intermédiaire avec d'autres composants de bases de données, en particulier les composants de connexion et de commit.	

Scénario : Filtrer et agréger des colonnes d'une table directement sur le SGBD

Le scénario suivant crée un Job ouvrant une connexion à une base de données MySQL et :

- instancie en partie les schémas d'une table de base de données (pour filtrer les colonnes),
- filtre deux colonnes dans la même table afin de ne garder que les données correspondant aux deux conditions de filtre,
- collecte les données des colonnes filtrées, regroupées par valeur(s) spécifique(s) et écrit les données agrégées dans une table cible de base de données.

Pour filtrer et agréger les colonnes d'une table de base de données :

- Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tMysqlConnection**, **tCombinedSQLInput**, **tCombinedSQLFilter**, **tCombinedSQLAggregate**, **tCombinedSQLOutput** et **tMysqlCommit**.
- Connectez les composants **tMysqlConnection**, **tCombinedSQLInput** et **tMysqlCommit** à l'aide de liens **OnSubjobOk**.
- Connectez les composants **tCombinedSQLInput**, **tCombinedSQLFilter**, **tCombinedSQLAggregate** et **tCombinedSQLOutput** à l'aide d'un lien **Combine**.

- Dans l'espace de modélisation graphique, double-cliquez sur le composant **tMysqlConnection** pour afficher sa vue **Component** et définir ses propriétés de base.
- Dans la vue **Basic settings**, configurez manuellement les informations de connexion à la base de données ou sélectionnez **Repository** dans la liste **Property Type** si votre connexion à la base de données a déjà été configurée et stockée sous le nœud **Metadata** de la vue **Repository**.

Pour plus d'informations concernant le stockage des métadonnées dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

- Dans l'espace de modélisation graphique, double-cliquez sur le composant **tCombinedSQLInput** pour afficher sa vue **Component** et définir ses propriétés de base.

- Dans le champ **Table**, saisissez le nom de la table source puis cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin de définir la structure des données.

Le schéma défini dans le composant **tCombinedSQLInput** peut être différent de celui de la table source, puisque vous pouvez instancier uniquement les colonnes souhaitées de la table source. Le **tCombinedSQLInput** joue également un rôle de filtre des colonnes.

Dans ce scénario, la table de la base de données source contient sept colonnes : *id*, *first_name*, *last_name*, *city*, *state*, *date_of_birth*, ainsi que *salary* et le composant **tCombinedSQLInput** instancie uniquement les quatre colonnes nécessaires à l'agrégation : *id*, *state*, *date_of_birth*, et *salary* de la table source.

- Dans l'espace de modélisation graphique, double-cliquez sur le composant **tCombinedSQLFilter** afin d'afficher sa vue **Component** et définir ses propriétés de base.

- Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent, ou configurez manuellement le schéma en sélectionnant **Built-in** dans la liste **Schema** et en cliquant sur le bouton **[...]** à côté du champ **Edit schema**.

Lorsque vous définissez la structure des données du composant **tCombinedSQLFilter**, le nom des colonnes apparaît automatiquement dans la liste **Input column** dans le tableau **Conditions**.

Dans ce scénario, le composant **tCombinedSQLFilter** instancie quatre colonnes : *id*, *state*, *date_of_birth*, et *salary*.

- Dans le tableau **Conditions**, configurez les paramètres d'entrée, les opérateurs et les valeurs attendues pour extraire les enregistrements correspondant aux critères spécifiés.

Dans ce scénario, le composant **tCombinedSQLFilter** filtre les colonnes *state* et *date_of_birth* dans la table source afin d'extraire uniquement les employés nés après le 19 octobre 1960, habitant dans les Etats *Utah*, *Ohio* et *Iowa*.

- Sélectionnez **And** dans la liste **Logical operator between conditions** afin d'appliquer les deux conditions à la fois. Vous pouvez également personnaliser les conditions en cochant la case **Use custom SQL** et en éditant les conditions dans le champ de texte.

- Dans l'espace de modélisation graphique, double-cliquez sur le composant **tCombinedSQLAggregate** afin d'afficher sa vue **Component** et définir ses propriétés de base.

- Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent, ou configurez manuellement le schéma en sélectionnant **Built-in** dans la liste **Schema** et en cliquant sur le bouton **[...]**.

Le composant **tCombinedSQLAggregate** instancie quatre colonnes : *id*, *state*, *date_of_birth*, et *salary*, provenant du composant précédent.

Le tableau **Group by** vous permet de définir les ensembles de données à traiter à partir d'une colonne définie, *State* dans cet exemple.

- Dans le tableau **Group by**, cliquez sur le bouton **[+]** pour ajouter une ligne.
- Dans la liste déroulante **Output column**, sélectionnez *State*. Cette colonne sera utilisée pour stocker les données filtrées de la colonne *State*.

Le tableau **Operations** vous permet de configurer le type d'agrégation à effectuer. La liste disponible dans la colonne **Output column** dépend du schéma que vous souhaitez écrire en sortie (via le composant **tCombinedSQLOutput**). Dans ce scénario, le but est de regrouper les employés selon l'Etat dans lequel il vivent, de calculer le nombre d'employés par Etat, les salaires moyen/minimum/maximum ainsi que de retourner la date de naissance de l'employé le plus âgé et de l'employé de plus jeune de chaque état.

- Dans le tableau **Operations**, cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) puis cliquez dans la liste **Output column** afin de sélectionner la colonne de sortie qui contiendra les données calculées.
- Cliquez dans la ligne **Function** et sélectionnez l'opération à effectuer.
- Dans l'espace de modélisation graphique, double-cliquez sur le **tCombinedSQLOutput** afin d'afficher sa vue **Component** et définir ses propriétés de base.

- Dans la liste **Database type**, sélectionnez la base de données correspondante.
- Dans la liste **Component List**, sélectionnez le composant de connexion correspondant si plus d'un composant de connexion est utilisé.
- Dans le champ **Table**, saisissez le nom de la table cible qui stockera les résultats agrégés.

Dans cet exemple, il n'est pas nécessaire de renseigner le champ **Schema**, puisque la base de données utilisée est Oracle.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir la structure des données de la table cible.

Dans ce scénario, le composant **tCombinedSQLOutput** instancie sept colonnes provenant du composant précédent dans le Job (**tCombinedSQLAggregate**) : *state*, *empl_count*, *avg_salary*, *min_salary*, *max_salary*, *oldest_empl* et *youngest_empl*.

- Dans l'espace de modélisation graphique, double-cliquez sur le **tCombinedSQLCommit** afin d'afficher sa vue **Component** et définir ses propriétés de base.
- Dans la liste **Component list**, sélectionnez le composant de connexion à la base de données correspondant si plusieurs ont été utilisés.
- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

Des lignes sont insérées dans une table composée de sept colonnes, *empl_by_state*, dans la base de données. La table affiche, par Etat défini, le nombre d'employés, les salaires moyen/minimum/maximum et retourne la date de naissance de l'employé le plus âgé et de l'employé le plus jeune de chaque état.

state	empl_count	avg_salary	min_salary	max_salary	oldest_empl	youngest_empl
Utah	3	10167	5087	16180	1961-10-19 02	1986-12-31 18:21
Ohio	3	10646	6269	19380	1968-10-19 02	1975-06-27 01:08
Iowa	2	13149	8118	18180	1972-10-15 04	1975-06-27 01:08

tCombinedSQLFilter

Propriétés du tCombinedSQLFilter

Famille de composant	ELT/CombinedSQL	
Fonction	Le composant **tCombinedSQLFilter** vous permet de modifier le schéma de la table source via le mapping des colonnes et de définir un filtre sur cette table. Ce composant peut donc filtrer des colonnes et des lignes en même temps. Il filtre en temps réel les données dans le SGBD lui-même.	
Objectif	Ce composant permet de filtrer des données en réorganisant, supprimant ou ajoutant des colonnes à partir de la table source et de filtrer la source de données à l'aide de conditions de filtre.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Logical operator between conditions	Sélectionnez l'opérateur logique à ajouter entre les conditions de filtre définies dans le tableau **Conditions**. Les deux opérateurs disponibles sont : **Or** et **And**.
	Conditions	Cliquez sur le bouton **[+]** pour ajouter autant de conditions que voulu. Les conditions sont exécutées les unes après les autres pour chaque ligne.
		Input Column : Sélectionnez la colonne à filtrer.
		Operator : Sélectionnez le type de clause WHERE : =, < >, >, <, >=, <=, **LIKE**, **IN**, **NOT IN**, et **EXIST IN**.
		Value : Saisissez les valeurs à utiliser dans la clause WHERE.
		Negate : Cochez cette case pour activer la condition contraire à celle définie.
	Use custom SQL	Cochez cette case afin de personnaliser une clause WHERE en éditant le code dans le champ **SQL Condition**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.

Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **Flow** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé en tant que composant intermédiaire avec d'autres composants de bases de données, en particulier les composants de connexion et de commit.

Scénario associé

Pour un scénario associé, consultez *Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD*.

tCombinedSQLInput

Propriétés du tCombinedSQLInput

Famille de composant	ELT/CombinedSQL	
Fonction	Le composant **tCombinedSQLInput** extrait des champs d'une table de base de données à partir de son schéma. Ce composant possède également des fonctionnalités de filtre puisque son schéma peut être différent de celui de la table de la base de données.	
Objectif	Le composant **tCombinedSQLInput** extrait des champ d'une table de base de données selon la définition de son schéma. Il passe ensuite la liste des champs au composant suivant via un lien de type **Combine**. Le schéma du **tCombinedSQLInput** peut être différent de celui de la table source de la base de données mais l'ordre des colonnes doit être le même.	
Basic settings	*Table*	Saisissez le nom de la table source de la base de données.
	Schema	Saisissez le nom du schéma de la table source. Renseignez ce champ si vous utilisez Oracle.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le du *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Add additional columns	Cette option vous permet d'effectuer des actions sur les colonnes, à l'exception des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
		Name : Saisissez le nom de la colonne du schéma devant être modifiée.
		SQL expression : Saisissez l'expression SQL à exécuter pour modifier les données dans la colonne correspondante.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **Flow** et retourne un entier.	

	QUERY : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé en tant que composant intermédiaire avec d'autres composants de bases de données, en particulier les composants de connexion et de commit.

Scénario associé

Pour un scénario associé, consultez *Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD*.

tCombinedSQLOutput

Propriétés du tCombinedSQLOutput

Famille de composant	ELT/CombinedSQL	
Fonction	Le composant **tCombinedSQLOutput** insère des enregistrements dans une table de base de données existante.	
Objectif	Le **tCombinedSQLOutput** insère des enregistrements du flux entrant dans une table de base de données existante.	
Basic settings	*Database Type*	Sélectionnez le type de base de données.
	Component list	Sélectionnez le composant de connexion à la base de données correspondant si plus d'un est utilisé dans le Job.
	Table	Saisissez le nom de la table cible de la base de données.
	Schema	Saisissez le nom du schéma de la table cible. Renseignez ce champ si vous utilisez Oracle.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Action on data	Sélectionnez **INSERT** dans la liste afin d'insérer les enregistrements du flux d'entrée dans la table cible de la base de données.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **Flow** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé en tant que composant intermédiaire avec d'autres composants de bases de données, en particulier les composants de connexion et de commit.

Scénario associé

Pour un scénario associé, consultez *Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD*.

tDB2Connection

Le composant **tDB2Connection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tDB2Connection*.

tELTGreenplumInput

Propriétés du tELTGreenplumInput

Les trois composants ELT Greenplum sont étroitement liés en ce qui concernent leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas Greenplum DB afin de générer des commandes Insert, dont des clauses, qui sont exécutés dans la table de sortie définie.

Famille de composants	ELT/Map/Greenplum	
Fonction	Fournit le schéma de table à utiliser pour la requête SQL à exécuter.	
Objectif	Le composant **tELTGreenplumInput** vous permet d'ajouter autant de tables d'entrée (input) que nécessaire pour les requêtes Insert les plus complexes.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit la nature le nombre de champs à traiter. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur duStudio Talend*.
	Default Table Name	Saisissez le nom de la table par défaut, entre guillemets doubles.
	Default Schema Name	Saisissez le nom du schéma par défaut, entre guillemets doubles.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	L'utilisation du composant **tELTGreenplumOutput** est étroitement liée à celle des composants **tELTGreenplumMap** et **tELTGreenplumInput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

Scénarios associés

Pour des exemples d'utilisation du composant **tELTGreenplumInput**, consultez :

- *Scénario: Mapper les données à l'aide d'une jointure implicite simple*
- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*
- *Scénario 2 : ELT utilisant une table Alias*

tELTGreenplumMap

Propriétés du tELTGreenplumMap

Les trois composants ELT Greenplum sont étroitement liés en ce qui concernent leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas Greenplum DB afin de générer des commandes Insert, dont des clauses, qui sont exécutés dans la table de sortie définie.

Famille de composant	ELT/Map/ Greenplum	
Fonction	Le composant **tELTGreenplumMap** permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.	
Objectif	Ce composant utilise les tables fournies en entrée pour alimenter les paramètres dans la commande mise en place. L'instruction peut inclure des joints internes ou externes qui peuvent être implémentés entre des tables ou entre une table et ses alias.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	ELT Greenplum Map Editor	L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.
	Style link	Sélectionnez le type d'affichage des liens. **Auto** : par défaut, les liens entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbe. **Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne. **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d'optimiser les performances.
	Property type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur de la base.

	Database	Nom de la base de données
	Username et *Password*	Données d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTGreenplumMap** est étroitement liée à celle des composants **tELTGreenplumInput** et **tELTGreenplumOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	

Ajouter des clauses WHERE et d'autres clauses

Vous pouvez également restreindre une requête Select à l'aide d'une clause WHERE et/ou à l'aide d'autres clauses telles que GROUP BY, ORDER BY, etc., en cliquant sur le bouton **Add filter row** situé en haut de la table de sortie dans l'éditeur de map.

Pour ajouter une restriction à l'aide d'une clause WHERE, cliquez sur le bouton **Add filter row** et sélectionnez **Add a WHERE clause** dans le menu qui s'ouvre.

Pour ajouter une restriction à l'aide de GROUP BY, ORDER BY, etc., cliquez sur le bouton **Add filter row** et sélectionnez **Add an other(GROUP...) clause** dans le menu qui s'ouvre.

Veillez à ce que les composants d'entrée soient tous liés correctement au composant ELT Map pour garantir que les inclusions, jointures et clauses fonctionnent.

Scénario: Mapper les données à l'aide d'une jointure implicite simple

Dans ce scénario, un composant **tELTGreenplumMap** est utilisé afin de récupérer les données de la table source *employee_by_statecode*, comparer sa colonne*statecode* à la table *statecode* et enfin mapper les colonnes désirées à partir des deux tables vers la table de sortie *employee_by_state*.

Avant l'exécution du Job, les trois tables *employee_by_statecode*, *statecode* et *employee_by_state* ressemblent à ceci :

id	name	statecode
1	Andy	86
3	Mike	39
2	John	52

state	statecode
China	86
Mexico	52
Italy	39

id	name	state

Déposer les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tGreenplumConnection**, deux **tELTGreenplumInput**, un **tELTGreenplumMap**, un**tELTGreenplumOutput**, un **tGreenplumCommit**, un**tGreenplumInput** et un **tLogRow**.

2. Renommez le composant **tGreenplumConnection** en **connect_to_greenplum_host**, les deux **tELTGreenplumInput** en **employee+statecode** et **statecode**, le **tELTGreenplumMap** en **match+map**, le **tELTGreenplumOutput** en **map_data_output**, le **tGreenplumCommit** en **commit_to_host**, le **tGreenplumInput** en **read_map_output_table** et le **tLogRow** en **show_map_data**

3. Reliez le composant **tGreenplumConnection** au **tELTGreenplumMap** à l'aide d'un lien **OnSubjobOk**.

   Reliez le composant **tELTGreenplumMap** au **tGreenplumCommit** à l'aide d'un lien **OnSubjobOk**.

   Reliez le composant **tGreenplumCommit** au **tGreenplumInput** à l'aide d'un lien **OnSubjobOk**.

4. Reliez le composant **tGreenplumInput** au **tLogRow** à l'aide d'une connexion **Row > Main**.

   Les deux **tELTGreenplumInput** et le composant **tELTGreenplumOutput** seront connectés plus tard au **tELTGreenplumMap**, une fois les tables correspondantes définies.

Configurer les composants

1. Double-cliquez sur le composant **tGreenplumConnection** pour ouvrir sa vue **Basic settings**.

   Dans les champs **Host** et **Port**, saisissez les variables de contexte pour le serveur Greenplum.

   Dans le champ **Database**, saisissez la variable de contexte pour la base de données Greenplum.

   Dans les champs **Username** et **Password**, saisissez les variables de contexte pour les paramètres d'authentification.

   Pour plus d'informations sur les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

2. Double-cliquez sur **employee+statecode** pour ouvrir sa vue **Basic settings** dans l'onglet **Component**.

   Dans le champ **Default table name**, saisissez le nom de la table source, nommée *employee_by_statecode*.

   Cliquez sur le bouton **[...]** près du champ **Edit schema** pour ouvrir l'éditeur de schéma.

Cliquez sur le bouton **[+]** pour ajouter trois colonnes et appelez-les *id*, *name* et *statecode*, avec pour type respectivement *INT4*, *VARCHAR*, et *INT4*.

Cliquez sur **OK** pour fermer l'éditeur de schéma.

Reliez le **employee+statecode** au **tELTGreenplumMap** en utilisant le composant de sortie *employee_by_statecode*.

3. Double-cliquez sur le **statecode** pour ouvrir sa vue **Basic settings** dans l'onglet **Component**.

Dans le champ **Default table name**, saisissez le nom de la table, ici *statecode*.

4. Cliquez sur le bouton **[...]** près du champ **Edit schema** pour ouvrir l'éditeur de schéma.

Cliquez sur le bouton **[+]** pour ajouter deux colonnes et appelez-les *state* et *statecode*, avec pour type respectivement *VARCHAR* et *INT4*.

Cliquez sur **OK** pour fermer l'éditeur de schéma.

Reliez le **statecode** au **tELTGreenplumMap** en utilisant la sortie *statecode*.

5. Cliquez le composant **tELTGreenplumMap** pour ouvrir sa vue **Basic settings** dans l'onglet **Component**.

Cochez la case **Use an existing connection**.

6. Cliquez sur le bouton **[...]** près du champ **ELT Greenplum Map Editor** pour ouvrir l'éditeur de mapping.

7. Cliquez sur le bouton **[+]** en haut à gauche pour ouvrir la fenêtre de sélection de table.

Sélectionnez les tables *employee_by_statecode* et *statecode* dans la liste et cliquez sur **Ok**.

Les tables apparaissent sur le panneau gauche de l'éditeur.

8. Dans le coin en haut à droite, cliquez sur le bouton **[+]** pour ajouter la table de sortie nommée *employee_by_state*.

   Cliquez sur **OK** pour fermer l'éditeur de map.

9. Double-cliquez sur le **tELTGreenplumOutput** pour ouvrir sa vue **Basic settings** dans l'onglet **Component**.

   Dans le champ **Default table name**, saisissez le nom de la table de sortie, ici *employee_by_state*.

10. Cliquez sur le bouton **[...]** près du champ **Edit schema** pour ouvrir l'éditeur de schéma.

Cliquez sur le bouton **[+]** pour ajouter trois colonnes et appelez-les respectivement *id*, *name* et *state*, avec pour type respectivement *INT4*, *VARCHAR*, et *VARCHAR* .

Cliquez sur **OK** pour fermer l'éditeur de schéma.

Reliez le composant **tELTGreenplumMap** au **tELTGreenplumOutput** en utilisant la table de sortie *employee_by_state*.

Cliquez sur **OK** dans la fenêtre pop-up pour récupérer le schéma du **tELTGreenplumOutput**.

La table de sortie *employee_by_state* partage maintenant le même schéma que celui du **tELTGreenplumOutput**.

11. Double-cliquez sur le composant **tELTGreenplumMap** pour ouvrir l'éditeur de map.

    Déposez la colonne *statecode* de la table *employee_by_statecode* dans la colonne du même nom, dans la table *statecode*, afin de chercher les enregistrements des deux tables ayant les mêmes valeurs *statecode*.

    Déposez les colonnes *id* et *name* de la table *employee_by_statecode* ainsi que la colonne *statecode* de la table *statecode* dans les colonnes de même nom dans la table de sortie *employee_by_state*.

    Cliquez sur **OK** pour fermer l'éditeur de mapping.

12. Double-cliquez sur le **tGreenplumInput** pour ouvrir sa vue **Basic settings** dans l'onglet **Component**.

    Cochez la case **Use an existing connection**.

    Dans le champ **Table name**, saisissez le nom de la table source, appelée *employee_by_state*.

    Dans le champ **Query**, saisissez la commande `"SELECT * FROM \"employee_by_state\""`.

13. Double-cliquez sur le composant **tLogRow** pour ouvrir sa vue **Basic settings** dans l'onglet **Component**.

Dans la zone **Mode**, sélectionnez l'option **Table (print values in cells of a table)** pour une meilleure lisibilité.

Exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** pour exécuter le Job.

```
--> 3 rows inserted.
.--+----+------.
|show_map_data  |
|=-+----+-----=|
|id|name|state  |
|=-+----+-----=|
|2 |John|Mexico|
|1 |Andy|China  |
|3 |Mike|Italy  |
'--+----+------'|

[statistics] disconnected
```

Comme affiché ci-dessus, les données sur les employés ont été écrites dans la table *employee_by_state*, qui présente des informations géographiques concernant les employés.

Scénario associé

Pour des scénarios associés, consultez les scénarios :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.
- Pour un scénario expliquant les sous-requêtes, consultez *Scénario : Mapper des données à l'aide d'une sous-requête*.

tELTGreenplumOutput

Propriétés du tELTGreenplumOutput

Les trois composants ELT Greenplum sont étroitement liés en ce qui concernent leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas Greenplum DB afin de générer des commandes Insert, dont des clauses, qui sont exécutés dans la table de sortie définie.

Famille du composant	ELT/Map/Greenplum	
Fonction	Exécute les actions sur la table spécifiée et insère les données selon le schéma de sortie défini dans le ELT Mapper.	
Objectif	Exécute les requêtes SQL Insert, Update et Delete dans la base de données Greenplum.	
Basic settings	*Action on data*	Sur les données de la table définie, vous pouvez effectuer les opérations suivantes : **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête. **Update** : Met à jour des lignes existantes. **Delete** : Supprime les entrées correspondantes au flux d'entrée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit la nature le nombre de champs à traiter. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur duStudio Talend*.
	Where clauses for (for UPDATE and DELETE only)	Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.
	Default Table Name	Saisissez le nom de la table par défaut, entre guillemets doubles.
	Default Schema Name	Saisissez le nom du schéma par défaut, entre guillemets doubles.
	Use different table name	Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ **Table name** qui apparaît.

Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTGreenplumOutput** est étroitement liée à celle des composants **tELTGreenplumMap** et **tELTGreenplumInput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour des exemples d'utilisation du composant **tELTGreenplumOutput**, consultez :

- *Scénario: Mapper les données à l'aide d'une jointure implicite simple*
- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*
- *Scénario 2 : ELT utilisant une table Alias*

tELTJDBCInput

Propriétés du tELTJDBCInput

Les composants **tELTJDBCInput**, **tELTJDBCOutput**, et **tELTJDBCMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base JDBC afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/JDBC	
Fonction	Fournit le schéma de la table d'entrée à la commande SQL qui sera exécutée.	
Objectif	Permet d'ajouter autant de tables que nécessaires dans une commande Insert qui peut être complexe.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l'utiliser dans la commande SQL Insert. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Default Table Name	Saisissez le nom de la table par défaut.
	Default Schema Name	Saisissez le nom du schéma par défaut.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	L'utilisation du composant **tELTJDBCInput** est étroitement liée à celle du **tELTJDBCMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

Scénarios associés

Pour des scénarios utilisant un composant semblable au **tELTJDBCInput**, consultez les scénarios suivants du composant **tELTMysqlMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTJDBCMap

Propriétés du tELTJDBCMap

Les composants **tELTJDBClInput**, **tELTJDBCOutput**, et **tELTJDBCMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base JDBC afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/JDBC	
Fonction	Permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.	
Objectif	Utilise les tables fournies en entrée pour alimenter les paramètres nécessaires à l'exécution de la commande SQL souhaitée. La commande peut contenir des jointures internes comme externes entre différentes tables ou entre une table et ses alias.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	ELT JDBC Map Editor	L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.
	Style link	Sélectionnez le type d'affichage des liens. **Auto** : par défaut, les liens entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbe. **Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne. **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d'optimiser les performances.
	Property type	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur de la base.

	Database	Nom de la base de données.
	Username et *Password*	Données d'authentification de l'utilisateur de la base. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTJDBCMap** est étroitement liée à celle des composants **tELTJDBCInput** et **tELTJDBCOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	

Ajouter des clauses WHERE et d'autres clauses

Vous pouvez également restreindre une requête Select à l'aide d'une clause WHERE et/ou à l'aide d'autres clauses telles que GROUP BY, ORDER BY, etc., en cliquant sur le bouton **Add filter row** situé en haut de la table de sortie dans l'éditeur de map.

Pour ajouter une restriction à l'aide d'une clause WHERE, cliquez sur le bouton **Add filter row** et sélectionnez **Add a WHERE clause** dans le menu qui s'ouvre.

Pour ajouter une restriction à l'aide de GROUP BY, ORDER BY, etc., cliquez sur le bouton **Add filter row** et sélectionnez **Add an other(GROUP...) clause** dans le menu qui s'ouvre.

Veillez à ce que les composants d'entrée soient tous liés correctement au composant ELT Map pour garantir que les inclusions, jointures et clauses fonctionnent.

Scénarios associés

Pour des scénarios associés, consultez les scénarios suivants :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.
- Pour un scénario expliquant les sous-requêtes, consultez *Scénario : Mapper des données à l'aide d'une sous-requête*.

tELTJDBCOutput

Propriétés du tELTJDBCOutput

Les composants **tELTJDBCInput**, **tELTJDBCOutput**, et **tELTJDBCMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base JDBC afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille du composant	ELT/Map/JDBC	
Fonction	Effectue l'action sur la table telle que spécifiée et insère les données selon le schéma de sortie défini dans le Mapper ELT.	
Objectif	Exécute la requête Insert dans la base de données JDBC.	
Basic settings	*Action on data*	Sur les données de la table définie, vous pouvez effectuer les opérations suivantes : **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête. **Update** : Met à jour des lignes existantes. **Delete** : Supprime les entrées correspondantes au flux d'entrée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Where clauses for (for UPDATE and DELETE only)	Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.
	Default Table Name	Saisissez le nom de la table par défaut, entre guillemets doubles.
	Default Schema Name	Saisissez le nom du schéma par défaut, entre guillemets doubles.

	Use different table name	Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ **Table name** qui apparaît.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTJDBCOutput** est étroitement liée à celle des composants **tELTJDBCInput** et **tELTJDBCMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour des scénarios utilisant un composant semblable au **tELTJDBCOutput**, consultez les scénarios suivants du composant **tELTJDBCMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTMSSqlInput

Propriétés du tELTMSSqlInput

Les composants **tELTMSSqlInput**, **tELTMSSqlOutput**, et **tELTMSSqlMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base MSSql afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/MSSql	
Fonction	Fournit le schéma de la table d'entrée à la commande SQL qui sera exécutée.	
Objectif	Permet d'ajouter autant de tables que nécessaires dans une commande Insert qui peut être complexe.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l'utiliser dans la commande SQL Insert Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Default Table Name	Saisissez le nom de la table par défaut.
	Default Schema Name	Saisissez le nom du schéma par défaut.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	L'utilisation du composant **tELTMSSqlInput** est étroitement liée à celle du **tELTMSSqlMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

Scénarios associés

Pour des scénarios utilisant un composant semblable au **tELTMSSqlInput**, consultez les scénarios suivants du composant **tELTMysqlMap** dans :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTMSSqlMap

Propriétés du tELTMSSqlMap

Les composants **tELTMSSqlInput**, **tELTMSSqlOutput**, et **tELTMSSqlMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base MSSql afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/MSSql	
Fonction	Permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.	
Objectif	Utilise les tables fournies en entrée pour alimenter les paramètres nécessaires à l'exécution de la commande SQL souhaitée. La commande peut contenir des jointures internes comme externes entre différentes tables ou entre une table et ses alias.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	ELT MSSql Map Editor	L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.
	Style link	Sélectionnez le type d'affichage des liens. **Auto** : par défaut, les liens entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbe. **Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne. **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d'optimiser les performances.
	Property type	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur de la base.
	Database	Nom de la base de données.
	Username et *Password*	Données d'authentification de l'utilisateur de la base.

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTMSSqlMap** est étroitement liée à celle des composants **tELTMSSqlInput** et **tELTMSSqlOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	

Ajouter des clauses WHERE et d'autres clauses

Vous pouvez également restreindre une requête Select à l'aide d'une clause WHERE et/ou à l'aide d'autres clauses telles que GROUP BY, ORDER BY, etc., en cliquant sur le bouton **Add filter row** situé en haut de la table de sortie dans l'éditeur de map.

Pour ajouter une restriction à l'aide d'une clause WHERE, cliquez sur le bouton **Add filter row** et sélectionnez **Add a WHERE clause** dans le menu qui s'ouvre.

Pour ajouter une restriction à l'aide de GROUP BY, ORDER BY, etc., cliquez sur le bouton **Add filter row** et sélectionnez **Add an other(GROUP...) clause** dans le menu qui s'ouvre.

Veillez à ce que les composants d'entrée soient tous liés correctement au composant ELT Map pour garantir que les inclusions, jointures et clauses fonctionnent.

Scénarios associés

Pour des scénarios associés, consultez les scénarios suivants :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.
- Pour un scénario expliquant les sous-requêtes, consultez *Scénario : Mapper des données à l'aide d'une sous-requête*.

tELTMSSqlOutput

Propriétés du tELTMSSqlOutput

Les composants **tELTMSSqlInput**, **tELTMSSqlOutput**, et **tELTMSSqlMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base MSSql afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille du composant	ELT/Map/MSSql	
Fonction	Effectue l'action sur la table telle que spécifiée et insère les données selon le schéma de sortie défini dans le Mapper ELT.	
Objectif	Exécute la requête Insert dans la base de données MSSql.	
Basic settings	*Action on data*	Sur les données de la table définie, vous pouvez effectuer les opérations suivantes : **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête. **Update** : Met à jour des lignes existantes. **Delete** : Supprime les entrées correspondantes au flux d'entrée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur duStudio Talend*.
	Where clauses for (for UPDATE and DELETE only)	Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.
	Default Table Name	Saisissez le nom de la table par défaut, entre guillemets doubles.
	Default Schema Name	Saisissez le nom du schéma par défaut, entre guillemets doubles.

	Use different table name	Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ **Table name** qui apparaît.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTMSSqlOutput** est étroitement liée à celle des composants **tELTMSSqlInput** et **tELTMSSqlMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour des scénarios utilisant un composant semblable au **tELTMSSqlOutput**, consultez les scénarios suivants du composant **tELTMysqlMap** dans :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTMysqlInput

Propriétés du tELTMysqlInput

Les composants **tELTMysqlInput**, **tELTMysqlOutput**, et **tELTMysqlMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base MySQL afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Mysql	
Fonction	Fournit le schéma de la table d'entrée à la commande SQL qui sera exécutée.	
Objectif	Permet d'ajouter autant de tables que nécessaires dans une commande Insert qui peut être complexe.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l'utiliser dans la commande SQL Insert. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Default Table Name	Saisissez le nom de la table par défaut, entre guillemets doubles.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	L'utilisation du composant **tELTMysqlInput** est étroitement liée à celle du **tELTMysqlMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

Scénarios associés

Pour des scénarios utilisant le composant **tELTMysqlInput**, consultez les scénarios suivants du composant **tELTMysqlMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTMysqlMap

Propriétés du tELTMysqlMap

Les composants **tELTMysqlInput**, **tELTMysqlOutput**, et **tELTMysqlMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base MySQL afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Mysql	
Fonction	Permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.	
Objectif	Utilise les tables fournies en entrée pour alimenter les paramètres nécessaires à l'exécution de la commande SQL souhaitée. La commande peut contenir des jointures internes comme externes entre différentes tables ou entre une table et ses alias.	
Basic settings	*DB Version*	Sélectionnez la version de MySQL que vous utilisez.
	Use an existing connection	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	ELT Mysql Map editor	L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.
	Style link	Sélectionnez le type d'affichage des liens. **Auto** : par défaut, les liens entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbe. **Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne. **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d'optimiser les performances.
	Property type	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur de la base.
	Database	Nom de la base de données.

	Username et *Password*	Données d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Preview	L'aperçu est un cliché des données traitées dans le Mapper. Il apparaît lorsque les propriétés du Mapper ont été complétées. La synchronisation de l'aperçu n'est disponible qu'après avoir enregistrer les modifications.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTMysqlMap** est étroitement liée à celle des composants **tELTMysqlInput** et **tELTMysqlOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	

Connecter les composants ELT

Les composants ELT ne traitent pas les données en tant que telles mais les informations du schéma descriptif de la table qui seront utilisées pour alimenter les paramètres de la requête SQL à exécuter.

Par conséquent la connexion requise pour relier ces différents composants est simplement un lien qui porte le nom de la table.

Le nom de la sortie que vous donnez à ce lien lors de sa création doit correspondre exactement au nom de la table car ce paramètre sera utilisé lors de la génération de la commande SQL.

Voir également le *Guide utilisateur du Studio Talend*.

Tables de jointure et de correspondance

Dans le Mapper ELT, vous pouvez sélectionner des colonnes spécifiques des schémas d'entrée et les ajouter dans le schéma de sortie.

- De la même manière que dans l'éditeur du Mapper ETL, simplement glissez et déposez le contenu du schéma d'entrée vers le schéma de sortie.

- Utilisez les touches **Ctrl** et **Maj** pour la sélection multiple des colonnes contiguës ou non.

Vous pouvez mettre en place des jointures explicites pour extraire des données variées de différentes tables.

- Cliquez sur la liste déroulante **Join** et sélectionnez une jointure explicite.
- Les jointures suivantes sont disponibles : **Inner Join**, **Left Outer Join**, **Right Outer Join** ou **Full Outer Join** et **Cross Join**.
- Par défaut la jointure interne **(Inner Join)** est sélectionnée.

Vous pouvez également créer des tables **Alias** afin de sélectionner des données variées dans une même table.

- Dans la zone Input, cliquez sur le bouton plus **[+]** pour créer un Alias.
- Définissez la table sur laquelle l'alias est basé.
- Saisissez un nom pour la table alias, de préférence veillez à ce qu'il soit différent de la table principale pour éviter les confusions.

Ajouter des clauses WHERE et d'autres clauses

Vous pouvez également restreindre une requête Select à l'aide d'une clause WHERE et/ou à l'aide d'autres clauses telles que GROUP BY, ORDER BY, etc., en cliquant sur le bouton **Add filter row** situé en haut de la table de sortie dans l'éditeur de map.

Pour ajouter une restriction à l'aide d'une clause WHERE, cliquez sur le bouton **Add filter row** et sélectionnez **Add a WHERE clause** dans le menu qui s'ouvre.

Pour ajouter une restriction à l'aide de GROUP BY, ORDER BY, etc., cliquez sur le bouton **Add filter row** et sélectionnez **Add an other(GROUP...) clause** dans le menu qui s'ouvre.

Veillez à ce que les composants d'entrée soient tous liés correctement au composant ELT Map pour garantir que les inclusions, jointures et clauses fonctionnent.

Générer la requête SQL

La mise en correspondance des éléments à partir des schémas d'entrée vers les schémas de sortie crée automatiquement la requête Select correspondante.

Les éventuelles clauses sont également prises en compte automatiquement.

Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre

Le scénario suivant décrit un Job rassemblant plusieurs schémas d'une table de base de données en entrée puis utilise une clause dans la requête SQL afin de filtrer les données en sortie.

- Cliquez et déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : trois **tELTMysqlIntput**, un **tELTMysqlMap**, et un **tELTMysqlOutput**.
- Double-cliquez sur le premier **tELTMysqlInput** afin d'afficher sa vue **Basic settings**.

- Dans la liste **Schema**, sélectionnez **Repository**, cliquez sur le bouton **[...]** à côté du champ **Edit schema**. Dans la boîte de dialogue **[Repository Content]**, sélectionnez votre connexion à la base de données ainsi que le schéma souhaité.

  Le nom du schéma sélectionné apparaît automatiquement dans le champ **Default Table Name**.

  Dans ce scénario, la connexion à la base de données est *Talend_MySQL* et le schéma pour le premier composant d'entrée est *owners*.

- Configurez les deuxième et troisième composants **tELTMysqlInput** de la même manière, mais sélectionnez *cars* et *resellers* respectivement comme nom de schéma.

Dans ce scénario, tous les schémas d'entrée sont stockés dans la zone **Metadata** du **Repository**, afin que vous les retrouviez facilement. Pour plus d'informations concernant les métadonnées, consultez le *Guide utilisateur du Studio Talend*.

Vous pouvez également sélectionner les trois composants d'entrée en déposant les schémas correspondants de la zone **Metadata** dans l'espace de modélisation graphique et en double-cliquant sur le composant **tELTMysqlInput** dans la boîte de dialogue **[Components]**. Cela vous permet d'éviter les étapes de nommage des composants et de définition de leur schéma.

- Reliez les trois composants **tELTMysqlInput** au composant **tELTMysqlMap** à l'aide de liens nommés strictement selon le nom des tables de la base de données : *owners*, *cars* et *resellers*.
- Reliez ensuite le composant **tELTMysqlMap** au **tELTMysqlOutput** et nommez le lien *agg_result*, qui est le nom de la table de la base de données dans laquelle vous allez sauvegarder le résultat de l'agrégation.
- Cliquez sur le composant **tELTMysqlMap** afin d'afficher sa vue **Basic settings** et configurer ses propriétés.

- Sélectionnez **Repository** dans la liste **Property Type** et sélectionnez la connexion à la base de données utilisée pour les composants d'entrée.

  Toutes les informations de la base de données sont automatiquement récupérées.

- Laissez les paramètres configurés par défaut
- Double-cliquez sur le composant **tELTMysqlMap** afin d'ouvrir le **Map Editor** pour mettre en place les jointures entre les tables d'entrée (Input) et configurer le flux de sortie.

- Ajoutez les tables d'entrée en cliquant sur le bouton **[+]** dans le coin supérieur gauche de l'éditeur et sélectionnez les noms des tables correspondants dans la boîte de dialogue **[Add a new alias]**.
- Glissez-déposez la colonne *ID_Owner* à partir de la table *owners* vers la colonne correspondante dans la table *cars*.
- Dans la table *cars*, cochez la case **Explicit Join** en face de *ID_Owners*.

  Une jointure **INNER JOIN**, le type de jointure par défaut, s'affiche dans la liste des jointures.
- Glissez la colonne *ID_Resellers* à partir de la table *cars* vers la colonne correspondante de la table *Resellers* pour mettre en place la seconde jointure. Définissez à nouveau une option de jointure **INNER JOIN**.
- Puis sélectionnez les colonnes à agréger dans la table de sortie, *agg_result*.
- Déposez les colonnes *ID_Owners*, *Name* et *ID_Insurance* de la table *owners* dans la table de sortie.
- Déposez les colonnes *Registration*, *Make* et *Color* de la table *cars* dans la table de sortie.
- Déposez les colonnes *Name_Reseller* et *City* de la table *resellers* dans la table de sortie.
- Avec les colonnes correspondantes sélectionnées, la mise en correspondance apparaît en jaune et les jointures en violet.
- Appliquez un filtre sur la table de sortie. Cliquez sur le bouton **Add filter row** en haut de la table de sortie afin d'afficher le champ textuel **Additional clauses**, déposez la colonne *City* de la table *resellers* dans le champ textuel puis définissez la clause WHERE : `resellers.City ='Augusta'`.

- Cliquez sur l'onglet **Generated SQL Select query** afin d'afficher l'instruction SQL correspondante.

- Cliquez **OK** pour sauvegarder le paramétrage de l'ELT Map.
- Double-cliquez sur le composant **tELTMysqlOutput** afin d'afficher sa vue **Basic settings**.

- Dans la liste **Action on data** sélectionnez l'action que vous souhaitez effectuer sur les données.
- Sélectionnez **Repository** dans la liste **Schema** et définissez le schéma de sortie de la même manière que pour les schémas d'entrée. Notez que le schéma de sortie doit être nommé comme la table dans laquelle vous souhaitez sauvegarder vos résultats d'aggrégation, *agg_result* dans ce scénario.

Vous pouvez également utiliser un schéma en mode **Built-In** et récupérer la structure du schéma du composant précédent. Cependant, vous devez être sûr d'avoir spécifié une table cible existant dans votre base de données, ayant la même structure de données.

- Laissez les autres paramètres configurés par défaut.
- Sauvegardez votre Job et appuyez sur la touche **F6** pour l'exécuter.

  Toutes les données sélectionnées sont insérées dans la table *agg_result* comme spécifié dans l'instruction SQL.

	ID_Owner	Name	ID_Insurance	Registration	Make	Color	Name_Reseller	City
	69	John ROOSEVELT	84	UDR 821	Peugeot	green	Bill GARFIELD	AUGUSTA
	11	Andrew COOLIDGE	104	ZYX 387	BMW	yellow	Ulysses POLK	AUGUSTA
	10	Martin ADAMS	167	ZFF 348	Audi	blue	Rutherford HARRISON	AUGUSTA
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Scénario 2 : ELT utilisant une table Alias

Le scénario suivant décrit un Job mappant des informations provenant des deux tables d'entrée et d'une table Alias servant de table d'entrée virtuelle, vers une table de sortie. La table *employees* contient l'ID des employés, le numéro de leur service, leur nom et l'ID de leur responsable. Ces responsables sont également considérés comme des employés et par conséquent inclus dans la table *employees*. La table *dept* contient les informations de service des employés. La table Alias récupère le nom des responsables de la table *employees*.

- Glissez et déposez de la **Palette** dans l'espace de modélisation graphique deux composants **tELTMysqlInput**, un **tELTMysqlMap** et un **tELTMysqlOutput**.

  Nommez-les comme vous le souhaitez afin de décrire au mieux leur fonctionnalité.

- Double-cliquez sur le premier composant **tELTMysqlInput** afin d'afficher sa vue **Basic settings** et configurer ses propriétés de base.

- Sélectionnez **Repository** dans la liste **Schema** puis définissez la connexion à la base de données ainsi que le schéma, en cliquant sur le bouton **[...]** à côté du champ **Edit Schema**.

  La connexion à la base de données est *Talend_MySQL* et le schéma du premier composant d'entrée est *employees*.

  Dans ce scénario, les schémas d'entrée sont stockés dans le **Repository**, sous le nœud **Metadata**, afin que vous puissiez les récupérer plus facilement. Pour plus d'informations concernant les métadonnées, consultez le *Guide utilisateur du Studio Talend*.

- Configurez le second composant **tELTMysqlInput** de la même manière, mais sélectionnez *dept* comme nom de schéma.

- Double-cliquez sur le composant **tELTMysqlOutput** afin d'afficher sa vue **Basic settings** et définir ses propriétés.

- Dans la liste **Action on data**, sélectionnez l'action que vous souhaitez effectuer, **Insert** dans ce scénario.
- Dans la liste **Schema**, sélectionnez **Repository** et définissez le schéma de sortie de la même manière que pour les schémas d'entrée. Dans ce scénario, sélectionnez *result* comme schéma de sortie, qui est le nom de la table de la base de données utilisée pour stocker le résultat du mapping.

  Le schéma de sortie contient toutes les colonnes des schémas d'entrée ainsi qu'une colonne *ManagerName*.
- Laissez les autres paramètres configurés par défaut.
- Reliez les deux composants **tELTMysqlInput** au **tELTMysqlMap** à l'aide de liens **Link** nommés strictement selon le nom des tables d'entrée, *employees* et *dept* dans ce scénario.
- Reliez le composant **tELTMysqlMap** au **tELTMysqlOutput** à l'aide d'un lien **Link**. Cliquez sur **Yes** (Oui) à l'ouverture de la boîte de dialogue pour permettre au Mapper ELT de récupérer la structure de la table de sortie à partir du schéma de sortie.
- Cliquez sur le composant **tELTMysqlMap** afin d'afficher sa vue **Basic settings** et configurer ses propriétés de base.

- Dans la liste **Property Type**, sélectionnez **Repository** puis sélectionnez la connexion à la base de données utilisée dans les composants d'entrée.

  Les informations de connexion à la base de données sont automatiquement récupérées.
- Laissez les autres paramètres configurés par défaut.
- Cliquez sur le bouton **[...]** à côté du champ **ELT Mysql Map Editor** ou double-cliquez sur le composant **tELTMysqlMap** dans l'espace de modélisation graphique afin d'ouvrir le ELT Map Editor.

  Le composant **tELTMysqlMap** est connecté au composant de sortie, la table de sortie est donc affichée dans la zone correspondante.
- Ajoutez les tables d'entrée *employees* et *dept* dans la zone d'entrée, en cliquant sur le bouton **[+]** et en sélectionnant le nom des tables correspondant dans la boîte de dialogue **[Add a new alias]**.
- Lorsque vous créez une table Alias à partir de la table *employees* en sélectionnant *employees* dans la liste **Select the table to use** et en saisissant *Managers* dans le champ **Type in a valid alias**, dans la boîte de dialogue **[Add a new alias]**.

- Déposez la colonne *DeptNo* de la table *employees* dans la table *dept*.
- Cochez la case **Explicit join** devant la colonne *DeptNo* de la table *dept* afin de configurer une jointure **Inner Join**.
- Déposez la colonne *ManagerId* de la table *employees* dans la colonne *ID* de la table *Managers*.
- Cochez la case **Explicit join** devant la colonne *ID* de la table *Managers* et sélectionnez **LEFT OUTER JOIN** dans la list **Join**, afin que les lignes de sortie puissent contenir des valeurs Null.

- Déposez toutes les colonnes de la table *employees* dans les colonnes correspondantes de la table de sortie.
- Déposez les colonnes *DeptName* et *Location* de la table *dept* dans les colonnes correspondantes de la table de sortie.
- Déposez la colonne *Name* de la table *Managers* dans la colonne *ManagerName* de la table de sortie.

result (table1)

Expression	Column
employees.ID	ID
employees.DeptNo	DeptNo
employees.Name	Name
employees.ManagerID	ManagerID
dept.DeptName	DeptName
dept.Location	Location
Managers.Name	ManagerName

- Cliquez sur l'onglet **Generated SQL Select query** pour afficher l'instruction SQL à exécuter.

Schema editor | Expression editor | Generated SQL Select query for "table1" output

```
"SELECT
employees.ID , employees.DeptNo , employees.Name , employees.ManagerID , dept.DeptName ,
 dept.Location , Managers.Name
FROM
 employees INNER JOIN  dept ON(  dept.DeptNo = employees.DeptNo  )
 LEFT OUTER JOIN  employees Managers ON(  Managers.ID = employees.ManagerID  )"
```

- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

  La table de sortie *result* de la base de données contient toutes les informations concernant les employés, y compris le nom de leurs responsables respectifs.

	ID	DeptNo	Name	ManagerID	DeptName	Location	ManagerName
	1	10	Alex	(NULL)	R&D	New York	(NULL)
	2	20	Peter	(NULL)	Accounting	Dallas	(NULL)
	3	10	Mark	1	R&D	New York	Alex
	4	10	Michael	1	R&D	New York	Alex
	5	20	Christophe	2	Accounting	Dallas	Peter
	6	10	Stephane	3	R&D	New York	Mark
	7	10	Cedric	3	R&D	New York	Mark
	8	10	Bill	4	R&D	New York	Michael
	9	20	Jack	2	Accounting	Dallas	Peter
	10	10	Andrews	4	R&D	New York	Michael
*	NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Scénario associé

Pour un scénario expliquant les sous-requêtes, consultez *Scénario : Mapper des données à l'aide d'une sous-requête*.

tELTMysqlOutput

Propriétés du tELTMysqlOutput

Les composants **tELTMysqlInput**, **tELTMysqlOutput**, et **tELTMysqlMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base MySQL afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille du composant	ELT/Map/Mysql	
Fonction	Effectue l'action sur la table telle que spécifiée et insère les données selon le schéma de sortie défini dans le Mapper ELT.	
Objectif	Exécute la requête Insert dans la base de données Mysql.	
Basic settings	*Action on data*	Sur les données de la table définie, vous pouvez effectuer les opérations suivantes : **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête. **Update** : Met à jour des lignes existantes. **Delete** : Supprime les entrées correspondantes au flux d'entrée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Where clauses for (for UPDATE and DELETE only)	Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.
	Default Table Name	Saisissez le nom de la table par défaut, entre guillemets doubles.
	Use different table name	Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ **Table name** qui apparaît.

Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	L'utilisation du composant **tELTMysqlOutput** est étroitement liée à celle des composants **tELTMysqlInput** et **tELTMysqlMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénarios associés

Pour des scénarios utilisant le composant **tELTMysqlOutput**, consultez les scénarios suivants du composant **tELTMysqlMap** dans :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTNetezzaInput

Propriétés du tELTNetezzaInput

Les composants **tELTNetezzalInput**, **tELTNetezzaOutput**, et **tELTNetezzaMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base MySQL afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Netezza	
Fonction	Fournit le schéma de la table d'entrée à la commande SQL qui sera exécutée.	
Objectif	Permet d'ajouter autant de tables que nécessaires dans une commande Insert qui peut être complexe.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit la nature le nombre de champs à traiter. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur duStudio Talend*.
	Default Table Name	Saisissez le nom de la table par défaut, entre guillemets doubles.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTNetezzaInput** est étroitement liée à celle des composants **tELTGreenplumMap** et **tELTGreenplumOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.	

		Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario: Mapper les données à l'aide d'une jointure implicite simple* .
- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTNetezzaMap

Propriétés du tELTNetezzaMap

Les trois composants ELT Netezza sont étroitement liés en ce qui concernent leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas Netezza DB afin de générer des commandes Insert, dont des clauses, qui sont exécutés dans la table de sortie définie.

Famille de composant	ELT/Map/Netezza	
Fonction	Le composant **tELTNetezzaMap** permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.	
Objectif	Ce composant utilise les tables fournies en entrée pour alimenter les paramètres dans la commande mise en place. L'instruction peut inclure des jointures internes ou externes qui peuvent être implémentés entre des tables ou entre une table et ses alias.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	ELT Netezza Map Editor	L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.
	Style link	Sélectionnez le type d'affichage des liens. **Auto** : par défaut, les liens entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbe. **Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne. **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d'optimiser les performances.
	Property type	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur de la base.
	Database	Nom de la base de données.
	Username et *Password*	Données d'authentification de l'utilisateur de la base de données.

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTNetezzaMap** est étroitement liée à celle des composants **tELTNetezzaInput** et **tELTNetezzaOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	

Ajouter des clauses WHERE et d'autres clauses

Vous pouvez également restreindre une requête Select à l'aide d'une clause WHERE et/ou à l'aide d'autres clauses telles que GROUP BY, ORDER BY, etc., en cliquant sur le bouton **Add filter row** situé en haut de la table de sortie dans l'éditeur de map.

Pour ajouter une restriction à l'aide d'une clause WHERE, cliquez sur le bouton **Add filter row** et sélectionnez **Add a WHERE clause** dans le menu qui s'ouvre.

Pour ajouter une restriction à l'aide de GROUP BY, ORDER BY, etc., cliquez sur le bouton **Add filter row** et sélectionnez **Add an other(GROUP...) clause** dans le menu qui s'ouvre.

Veillez à ce que les composants d'entrée soient tous liés correctement au composant ELT Map pour garantir que les inclusions, jointures et clauses fonctionnent.

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario: Mapper les données à l'aide d'une jointure implicite simple* .
- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.
- Pour un scénario expliquant les sous-requêtes, consultez *Scénario : Mapper des données à l'aide d'une sous-requête*.

tELTNetezzaOutput

Propriétés du tELTNetezzaOutput

Les trois composants ELT Netezza sont étroitement liés en ce qui concernent leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas Netezza DB afin de générer des commandes Insert, dont des clauses, qui sont exécutés dans la table de sortie définie.

Famille de composant	ELT/Map/Netezza	
Fonction	Le **tELTNetezzaMap** exécute les actions sur la table spécifiée et insère les données selon le schéma de sortie défini dans le ELT Mapper.	
Objectif	Exécute les requêtes SQL Insert, Update et Delete dans la base de données Netezza.	
Basic settings	*Action on data*	Sur les données de la table définie, vous pouvez effectuer les opérations suivantes : **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête. **Update** : Met à jour des lignes existantes. **Delete** : Supprime les entrées correspondantes au flux d'entrée.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit la nature le nombre de champs à traiter. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent dans le Job.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur duStudio Talend*.
	Where clauses for (for UPDATE and DELETE only)	Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.
	Default Table Name	Saisissez le nom de la table par défaut, entre guillemets doubles.

	Use different table name	Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ **Table name** qui apparaît.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTNetezzaOutput** est étroitement liée à celle des composants **tELTNetezzaMap** et **tELTNetezzaInput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario: Mapper les données à l'aide d'une jointure implicite simple* .
- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTOracleInput

Propriétés du tELTOracleInput

Les composants **tELTOracleInput**, **tELTOracleOutput**, et **tELTOracleMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Oracle afin de générer des commandes Insert, Update ou Delete y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Oracle	
Fonction	Fournit le schéma de la table d'entrée à la commande SQL qui sera exécutée.	
Objectif	Permet d'ajouter autant de tables que nécessaires dans une commande Insert qui peut être complexe.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l'utiliser dans la commande SQL Insert. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Default Table Name	Saisissez le nom de la table par défaut, entre guillemets doubles.
	Default Schema Name	Saisissez le nom du schéma par défaut, entre guillemets doubles.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	L'utilisation du composant **tELTOracleInput** est étroitement liée à celle du **tELTOracleMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

Scénario associé

Pour un scénario utilisant le composant **tELTOracleInput**, consultez *Scénario : Mettre à jour les lignes d'une base Oracle* du composant **tELTOracleMap**.

tELTOracleMap

Propriétés du tELTOracleMap

Les composants **tELTOracleInput**, **tELTOracleOutput**, et **tELTOracleMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Oracle afin de générer des commandes Insert, Update ou Delete y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Oracle	
Fonction	Permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.	
Objectif	Utilise les tables fournies en entrée pour alimenter les paramètres nécessaires à l'exécution de la commande SQL souhaitée. La commande peut contenir des jointures internes comme externes entre différentes tables ou entre une table et ses alias.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	ELT Oracle Map Editor	L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.
	Style link	Sélectionnez le type d'affichage des liens. **Auto** : par défaut, les liens entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbe. **Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne. **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d'optimiser les performances.
	Property type	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Connection type	Liste déroulante des pilotes disponibles.
	DB Version	Sélectionnez la version d'Oracle que vous utilisez.

	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur de la base.
	Database	Nom de la base de données.
	Username et *Password*	Données d'authentification de l'utilisateur de la base. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*Additional JDBC Parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	Use Hint Options	Cochez cette case afin d'activer la zone de configuration des hints, afin de vous permettre d'optimiser l'exécution d'une requête. Dans cette zone, les paramètres sont : -**HINT** : spécifiez le hint dont vous avez besoin, à l'aide de la syntaxe `/*+ */`. -**POSITION** : spécifiez où placer le hint dans l'instruction SQL. -**SQL STMT** : sélectionnez l'instruction SQL à utiliser.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTOracleMap** est étroitement liée à celle des composants **tELTOracleInput** et **tELTOracleOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	

Connecter les composants ELT

Pour plus d'informations concernant la connexion des composants ELT, consultez *Connecter les composants ELT*

Voir également le *Guide utilisateur du Studio Talend*.

Tables de jointure et de correspondance

Dans le Mapper ELT, vous pouvez sélectionner des colonnes des schémas d'entrée et les intégrer dans le schéma de sortie.

Pour plus d'informations concernant la mise en correspondance et les jointures, consultez *Tables de jointure et de correspondance*.

Lorsque vous devez faire une jointure entre de nombreuses tables, ou faire une jointure entre tables avec des conditions multiples et des jointures externes, il est recommandé d'utiliser les options **LEFT OUTER JOIN (+)** et **RIGHT OUTER JOIN (+)** qui vous permettent d'utiliser les mots clé privés Oracle. Pour plus d'informations concernant ces deux mots clé privés, consultez le site (en anglais) : http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/queries006.htm

Ajouter des clauses WHERE et d'autres clauses

Vous pouvez également restreindre une requête Select à l'aide d'une clause WHERE et/ou à l'aide d'autres clauses telles que GROUP BY, ORDER BY, etc., en cliquant sur le bouton **Add filter row** situé en haut de la table de sortie dans l'éditeur de map.

Pour ajouter une restriction à l'aide d'une clause WHERE, cliquez sur le bouton **Add filter row** et sélectionnez **Add a WHERE clause** dans le menu qui s'ouvre.

Pour ajouter une restriction à l'aide de GROUP BY, ORDER BY, etc., cliquez sur le bouton **Add filter row** et sélectionnez **Add an other(GROUP...) clause** dans le menu qui s'ouvre.

Veillez à ce que les composants d'entrée soient tous liés correctement au composant ELT Map pour garantir que les inclusions, jointures et clauses fonctionnent.

Générer la requête SQL

La mise en correspondance d'éléments des schémas d'entrée avec le schéma de sortie crée automatiquement la requête Select correspondante.

La/les clauses définies dans le mapper ELT sont également intégrées automatiquement dans la requête.

Scénario : Mettre à jour les lignes d'une base Oracle

Le scénario suivant est basé sur le scénario d'agrégation, dans *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*. L'action sur les données Update étant disponible en base Oracle, le scénario suivant décrit un Job qui met à jour des données particulières dans la table *agg_result*.

- Comme décrit dans *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*, créez un Job d'agrégation de données utilisant les composants ELT pour la base de données Oracle : **tELTOracleInput**, **tELTOracleMap** et **tELTOracleOutput**, puis exécutez le Job afin de sauvegarder le résultat de l'agrégation dans une table de base de données nommée *agg_result*.

Lorsque vous définissez des filtres dans l'éditeur ELT Map, notez que les chaînes de caractères sont sensibles à la casse dans la base de données Oracle.

- Lancez l'éditeur ELT Map et ajoutez une nouvelle table de sortie nommée *update_data*.
- Ajoutez une ligne de filtre à la table *update_data* pour établir une relation entre tables d'entrée et de sortie : `owners.ID_OWNER = agg_result.ID_OWNER`.
- Glissez la colonne *MAKE* de la table cars à la table *update_data*.
- Glissez la colonne *NAME_RESELLER* de la table *resellers* à la table *update_data*.
- Ajoutez un modèle entouré de guillemets simples, *A8* dans cet exemple, dans la colonne *MAKE* de la table *cars*, précédé de deux barres verticales.
- Ajoutez *Sold by*, entouré de guillemets simples devant la colonne *Name_Reseller* dans la table *resellers*, avec deux barres verticales au milieu.

- Vérifiez la requête Select générée dans l'onglet **Generated SQL select query**.

- Cliquez sur **OK** pour valider les modifications et fermer le mapper ELT.
- Désactivez le composant **tELTOracleOutput** nommé *Agg_Result* en cliquant-droit dessus et en sélectionnant **Deactivate Agg_Result** dans le menu contextuel.
- Déposez un nouveau composant **tELTOracleOutput** de la **Palette** dans l'espace de modélisation graphique. Nommez-le *Update_Data* afin d'identifier plus clairement sa fonction.
- Reliez le composant **tELTOracleMap** au nouveau **tELTOracleOutput** à l'aide du lien correspondant à la nouvelle table de sortie définie dans le mapper, *update_data* dans ce scénario.
- Double-cliquez sur le composant **tELTOracleOutput** afin d'afficher sa vue **Component** et définir ses propriétés.

- Sélectionnez **Update** dans la liste **Action on data**.
- Vérifiez le schéma et cliquez sur **Sync columns** pour récupérer la structure du schéma du composant précédent.
- Dans le champ **WHERE clauses**, entrez la clause suivante : `agg_results.MAKE = 'Audi'`, pour mettre à jour les données relatives à la marque Audi dans la table de la base de données *agg_result*.
- Renseignez le champ **Default Table Name** en saisissant le nom du lien de sortie, *update_data* dans cet exemple.
- Cochez la case **Use different table name** et renseignez le champ **Table name** en saisissant le nom de la table à mettre à jour, *agg_result* dans ce scénario.
- Laissez les autres paramètres tels qu'ils sont.
- Sauvegardez votre Job puis appuyez sur **F6** pour l'exécuter.

  Les données concernées ont été mises à jour dans la table de la base de données.

ID_OW...	NAME	ID_INS...	REGISTR...	MAKE	COLOR	NAME_RESELLER	CITY
10	Martin ADAMS	167	ZFF 348	Audi A8	grey	Sold by Rutherford HARRISON	AUGUSTA
11	Andrew COOLIDGE	104	ZYX 387	BMW	yellow	Ulysses POLK	AUGUSTA
69	John ROOSEVELT	84	UDR 821	Peugeot	green	Bill GARFIELD	AUGUSTA

Scénario associé

Pour un scénario expliquant les sous-requêtes, consultez *Scénario : Mapper des données à l'aide d'une sous-requête*.

tELTOracleOutput

Propriétés du tELTOracleOutput

Les composants **tELTOracleInput**, **tELTOracleOutput**, et **tELTOracleMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Oracle afin de générer des commandes Insert, Update ou Delete y compris des clauses, et ces commandes devront être exécutées dans la table de sortie définie.

Famille du composant	ELT/Map/Oracle	
Fonction	Effectue l'action sur la table telle que spécifiée et insère les données selon le schéma de sortie défini dans le Mapper ELT.	
Objectif	Exécute la requête Insert ou Update dans la base de données Oracle.	
Basic settings	*Action on data*	Sur les données de la table définie, vous pouvez effectuer les opérations suivantes : **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête. **Update** : Met à jour des lignes existantes. **Delete** : Supprime les entrées correspondantes au flux d'entrée. **MERGE** : met à jour et/ou insère les données dans la table. Notez que les options disponibles pour les opérations de MERGE sont différentes de celles disponibles pour les opérations d'Insert, d'Update ou de Delete. Les variables globales suivantes sont disponibles : • NB_LINE_INSERTED : Nombre de lignes insérées pendant l'opération d'**Insert**. • NB_LINE_UPDATED : Nombre de lignes mises à jour pendant l'opération d'**Update**. • NB_LINE_DELETED : Nombre de lignes supprimées pendant l'opération de **Delete**. • NB_LINE_MERGED : Nombre de lignes insérées ou mises à jour pendant l'opération de **MERGE**.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous

		souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Where clauses for (for UPDATE and DELETE only)	Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.
	Use Merge Update (pour **MERGE**)	Cochez cette case pour mettre à jour des données de votre table de sortie. **Column** : Liste des colonnes du flux d'entrée. **Update** : Cochez la case correspondant au nom de la colonne que vous souhaitez mettre à jour. **Use Merge Update Where Clause** : Cochez cette case et saisissez la clause WHERE permettant de filtrer les données à mettre à jour, si nécessaire. **Use Merge Update Delete Clause** : Cochez cette case et saisissez la clause WHERE permettant de filtrer les données à supprimer et mettre à jour, si nécessaire.
	Use Merge Insert (pour **MERGE**)	Cochez cette case pour insérer des données dans la table. **Column** : Liste des colonnes du flux d'entrée. **Check All** : Cochez la case correspondant au nom de la colonne que vous souhaitez insérer. **Use Merge Update Where Clause** : Cochez cette case et saisissez la clause WHERE permettant de filtrer les données à insérer.
	Default Table Name	Saisissez le nom de la table par défaut.
	Default Schema Name	Saisissez le nom du schéma Oracle par défaut.
	Use different table name	Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ **Table name** qui apparaît.
Advanced settings	*Use Hint Options*	Cochez cette case pour activer la zone de configuration des hints lorsque vous souhaitez utiliser un hint pour optimiser l'exécution d'une requête. Dans cette zone, les paramètres sont : - **HINT** : spécifiez l'hint dont vous avez besoin dans une instruction SQL. - **POSITION** : spécifiez où mettre l'hint dans une instruction SQL. - **SQL STMT** : sélectionnez l'instruction SQL dont vous avez besoin.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	L'utilisation du composant **tELTOracleOutput** est étroitement liée à celle des composants **tELTOracleInput** et **tELTOracleMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Utiliser la fonction MERGE d'Oracle pour mettre à jour et insérer simultanément des données

Ce scénario décrit un Job vous permettant d'ajouter de nouvelles informations client et de mettre à jour les informations client existantes dans une table de base de données à l'aide de la commande **MERGE** d'Oracle.

Relier les composants

1. De la **Palette**, glissez-déposez les composants suivants dans l'espace de modélisation graphique : **tELTOracleInput**, **tELTOracleMap** ainsi qu'un **tELTOracleOutput**.

2. Renommez le **tELTOracleInput** en **new_customer**, le **tELTOracleMap** en **ELT Mapper** et le **tELTOracleOutput** en **merge_data**.

3. Reliez le **tELTOracleInput** au **tELTOracleMap** à l'aide d'un lien **Row** > **New Output (table)**.

   Dans la boîte de dialogue qui s'ouvre, saisissez le nom de la table, *NEW_CUSTOMERS*. Ce nom doit correspondre au nom de la table de la base de données existante.

4. Reliez le **tELTOracleMap** au **tELTOracleOutput** à l'aide d'un lien **Row** > **New Output (table)**.

   Dans la boîte de dialogue qui s'ouvre, saisissez le nom de la table de la base de données, *customers_merge*, qui recevra les résultats de l'opération de **MERGE**.

Configurer les composants

1. Double-cliquez sur le **tELTOracleInput** afin d'afficher sa vue **Basic settings**.

2. Dans la liste **Schema**, sélectionnez **Repository**, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** puis sélectionnez votre connexion de base de données ainsi que le schéma souhaité dans la boîte de dialogue **[Repository Content]**.

Le nom du schéma sélectionné apparaît automatiquement dans le champ **Default Table Name**.

Dans ce scénario, la connexion à la base de données est *Talend_Oracle* et le schéma est *new_customers*.

Dans ce scénario, le schéma d'entrée est stocké sous le nœud **Metadata** du **Repository** afin de le retrouver facilement. Pour plus d'informations concernant les métadonnées, consultez le *Guide utilisateur du Studio Talend*.

Vous pouvez également sélectionner le composant d'entrée en déposant le schéma correspondant de la zone **Metadata** dans l'espace de modélisation graphique et en double-cliquant sur le **tELTOracleInput** dans la boîte de dialogue **[Components]**. Cela vous permet d'éviter les étapes de nommage du composant d'entrée ainsi que la définition manuelle de son schéma.

3. Cliquez sur le composant **tELTOracleMap** afin d'afficher sa vue **Basic settings**.

4. Sélectionnez **Repository** dans la liste **Property Type** et choisissez la connexion à la base de données que vous avez utilisée pour les composants d'entrée.

Les informations relatives à la base de données sont automatiquement récupérées.

Laissez les autres champs tels qu'ils sont.

5. Double-cliquez sur le composant **tELTOracleMap** afin d'ouvrir l'**ELT Map Editor** et configurer le flux de transformation de données.

   Affichez la table d'entrée en cliquant sur le bouton **[+]** dans le coin supérieur gauche de l'**ELT Map Editor** et en sélectionnant le nom de la table correspondante dans le boîte de dialogue **[Add a new alias]**.

   Dans ce scénario, la seule table d'entrée est *new_customers*.

6. Sélectionnez toutes les colonnes de la table d'entrée et déposez-les dans la table de sortie.

7. Cliquez sur **Generated SQL Select query** pour afficher l'instruction de la requête à exécuter.

   Cliquez sur **OK** pour valider les paramètres **ELT Map** et fermer l'**ELT Map Editor**.

8. Dans l'espace de modélisation, double-cliquez sur le composant **tELTOracleOutput** pour afficher sa vue **Basic settings**.

   Dans la liste **Action on data**, sélectionnez **MERGE**.

   Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.

Cochez la case **Use Merge Update** pour mettre à jour les données via la fonction MERGE d'Oracle.

9. Dans le tableau qui apparaît, cochez les cases correspondant aux colonnes que vous souhaitez mettre à jour. L'objectif est de mettre à jour les données client en fonction de leur ID. Ainsi, cochez toutes les cases sauf celle correspondant à la colonne *ID*.

Les colonnes définies comme clé primaire ne peuvent pas et ne doivent pas faire l'objet d'une mise à jour.

10. Cochez la case **Use Merge Insert** afin d'insérer de nouvelles données lors de la mise à jour des données existantes en utilisant la fonction MERGE d'Oracle.

    Dans le tableau qui apparaît, cochez la case des colonnes dans lesquelles vous souhaitez insérer de nouvelles données.

    Dans ce scénario, vous allez insérer de nouvelles données client. Cochez donc toutes les cases en cliquant sur la case **Check All**.

11. Renseignez le champ **Default Table Name** en saisissant le nom de la table cible déjà existante dans votre base de données. Dans cet exemple, saisissez *customers_merge*.

12. Laissez les autres paramètres tels qu'ils sont.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.

   Les données sont mises à jour et insérées dans la base de données. La requête utilisée est affichée dans la console.

```
[statistics] connecting to socket on port 3712
[statistics] connected
Merge with :
MERGE INTO customers_merge target USING (SELECT NEW_CUSTOMERS.ID ,
NEW_CUSTOMERS.FIRSTNAME , NEW_CUSTOMERS.LASTNAME , NEW_CUSTOMERS.ADDRESS ,
NEW_CUSTOMERS.REGISTERDATE , NEW_CUSTOMERS.REVENUE , NEW_CUSTOMERS.STATE  FROM
NEW_CUSTOMERS) source ON (target.ID=source.ID) WHEN MATCHED THEN UPDATE SET
target.FIRSTNAME=source.FIRSTNAME,target.LASTNAME=source.LASTNAME,target.ADDRESS=s
ource.ADDRESS,target.REGISTERDATE=source.REGISTERDATE,target.REVENUE=source.REVENU
E,target.STATE=source.STATE  WHEN NOT MATCHED THEN INSERT (
ID,FIRSTNAME,LASTNAME,ADDRESS,REGISTERDATE,REVENUE,STATE) VALUES (
source.ID,source.FIRSTNAME,source.LASTNAME,source.ADDRESS,source.REGISTERDATE,sour
ce.REVENUE,source.STATE)
Merge with :
MERGE INTO customers_merge target USING (SELECT NEW_CUSTOMERS.ID ,
NEW_CUSTOMERS.FIRSTNAME , NEW_CUSTOMERS.LASTNAME , NEW_CUSTOMERS.ADDRESS ,
NEW_CUSTOMERS.REGISTERDATE , NEW_CUSTOMERS.REVENUE , NEW_CUSTOMERS.STATE  FROM
NEW_CUSTOMERS) source ON (target.ID=source.ID) WHEN MATCHED THEN UPDATE SET
target.FIRSTNAME=source.FIRSTNAME,target.LASTNAME=source.LASTNAME,target.ADDRESS=s
ource.ADDRESS,target.REGISTERDATE=source.REGISTERDATE,target.REVENUE=source.REVENU
E,target.STATE=source.STATE  WHEN NOT MATCHED THEN INSERT (
ID,FIRSTNAME,LASTNAME,ADDRESS,REGISTERDATE,REVENUE,STATE) VALUES (
source.ID,source.FIRSTNAME,source.LASTNAME,source.ADDRESS,source.REGISTERDATE,sour
ce.REVENUE,source.STATE)

[statistics] disconnected
```

tELTPostgresqlInput

Propriétés du tELTPostgresqlInput

Les composants **tELTPostgresqlInput**, **tELTPostgresqlOutput**, et **tELTPostgresqlMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Postgresql afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Postgresql	
Fonction	Fournit le schéma de la table d'entrée à la commande SQL qui sera exécutée.	
Objectif	Permet d'ajouter autant de tables que nécessaires dans une commande Insert qui peut être complexe.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l'utiliser dans la commande SQL Insert. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Default Table Name	Saisissez le nom de la table par défaut.
	Default Schema Name	Saisissez le nom du schéma par défaut.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTPostgresqlInput** est étroitement liée à celle du **tELTPostrgesqlMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.	

		Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

Scénarios associés

Pour des scénarios utilisant un composant semblable au **tELTPostgresqlInput**, consultez les scénarios suivants du composant **tELTPostgresqlMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTPostgresqlMap

Propriétés du tELTPostgresqlMap

Les composants **tELTPostgresqlInput**, **tELTPostgresqlOutput** et **tELTPostgresqlMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Postgresql afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Postgresql	
Fonction	Permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.	
Objectif	Utilise les tables fournies en entrée pour alimenter les paramètres nécessaires à l'exécution de la commande SQL souhaitée. La commande peut contenir des jointures internes comme externes entre différentes tables ou entre une table et ses alias.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	ELT Postgresql Map Editor	L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.
	Style link	Sélectionnez le type d'affichage des liens. **Auto** : par défaut, les liens entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbe. **Curves** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne. **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d'optimiser les performances.
	Property type	Peut être Built-in ou Repository.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur de la base.
	Database	Nom de la base de données.
	Username et *Password*	Données d'authentification de l'utilisateur de la base de données.

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*Additional JDBC parameters*	Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n'est pas disponible lorsque vous utilisez l'option **Use an existing connection** dans les **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTPostgresqlMap** est étroitement liée à celle des composants **tELTPostgresqlInput** et **tELTPostgresqlOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	

Ajouter des clauses WHERE et d'autres clauses

Vous pouvez également restreindre une requête Select à l'aide d'une clause WHERE et/ou à l'aide d'autres clauses telles que GROUP BY, ORDER BY, etc., en cliquant sur le bouton **Add filter row** situé en haut de la table de sortie dans l'éditeur de map.

Pour ajouter une restriction à l'aide d'une clause WHERE, cliquez sur le bouton **Add filter row** et sélectionnez **Add a WHERE clause** dans le menu qui s'ouvre.

Pour ajouter une restriction à l'aide de GROUP BY, ORDER BY, etc., cliquez sur le bouton **Add filter row** et sélectionnez **Add an other(GROUP...) clause** dans le menu qui s'ouvre.

Veillez à ce que les composants d'entrée soient tous liés correctement au composant ELT Map pour garantir que les inclusions, jointures et clauses fonctionnent.

Scénarios associés

Pour des scénarios associés, consultez les scénarios suivants :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.
- Pour un scénario expliquant les sous-requêtes, consultez *Scénario : Mapper des données à l'aide d'une sous-requête*.

tELTPostgresqlOutput

Propriétés du tELTPostgresqlOutput

Les composants **tELTPostgresqlInput**, **tELTPostgresqlOutput**, et **tELTPostgresqlMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Postgresql afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille du composant	ELT/Map/Postgresql	
Fonction	Effectue l'action sur la table telle que spécifiée et insère les données selon le schéma de sortie défini dans le Mapper ELT.	
Objectif	Exécute la requête Insert dans la base de données Postgresql.	
Basic settings	*Action on data*	Sur les données de la table définie, vous pouvez effectuer les opérations suivantes : **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête. **Update** : Met à jour des lignes existantes. **Delete** : Supprime les entrées correspondantes au flux d'entrée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Where clauses for (for UPDATE and DELETE only)	Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.
	Default Table Name	Saisissez le nom de la table par défaut.
	Default Schema Name	Saisissez le nom du schéma par défaut.

	Use different table name	Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ **Table name** qui apparaît.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTPostgresqlOutput** est étroitement liée à celle des composants **tELTPostgresqlInput** et **tELTPostgresqlMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour des scénarios utilisant un composant semblable au **tELTPostgresqlOutput**, consultez les scénarios suivants du composant **tELTMysqlMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTSybaseInput

Propriétés du tELTSybaseInput

Les composants **tELTSybaseInput**, **tELTSybaseOutput**, et **tELTSybaseMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Sybase afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Sybase	
Fonction	Fournit le schéma de la table d'entrée à la commande SQL qui sera exécutée.	
Objectif	Permet d'ajouter autant de tables que nécessaires dans une commande Insert qui peut être complexe.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l'utiliser dans la commande SQL Insert Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Default Table Name	Saisissez le nom de la table par défaut.
	Default Schema Name	Saisissez le nom du schéma Sybase par défaut.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	L'utilisation du composant **tELTSybaseInput** est étroitement liée à celle du **tELTSybaseMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour des scénarios utilisant le composant **tELTSybaseInput**, consultez les scénarios suivants du composant **tELTMysqlMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTSybaseMap

Propriétés du tELTSybaseMap

Les composants **tELTSybaseInput**, **tELTSybaseOutput**, et **tELTSybaseMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Sybase afin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Sybase	
Fonction	Permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.	
Objectif	Utilise les tables fournies en entrée pour alimenter les paramètres nécessaires à l'exécution de la commande SQL souhaitée. La commande peut contenir des jointures internes comme externes entre différentes tables ou entre une table et ses alias.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	ELT Sybase Map Editor	L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.
	Style link	Sélectionnez le type d'affichage des liens. **Auto** : par défaut, les liens entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbe. **Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne. **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d'optimiser les performances.
	Property type	Peut être Built-in ou Repository.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Port	Numéro de port d'écoute du serveur de la base de données.
	Database	Nom de la base de données.
	Username et *Password*	Données d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.

Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTSybaseMap** est étroitement liée à celle des composants **tELTSybaseInput** et **tELTSybaseOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Ajouter des clauses WHERE et d'autres clauses

Vous pouvez également restreindre une requête Select à l'aide d'une clause WHERE et/ou à l'aide d'autres clauses telles que GROUP BY, ORDER BY, etc., en cliquant sur le bouton **Add filter row** situé en haut de la table de sortie dans l'éditeur de map.

Pour ajouter une restriction à l'aide d'une clause WHERE, cliquez sur le bouton **Add filter row** et sélectionnez **Add a WHERE clause** dans le menu qui s'ouvre.

Pour ajouter une restriction à l'aide de GROUP BY, ORDER BY, etc., cliquez sur le bouton **Add filter row** et sélectionnez **Add an other(GROUP...) clause** dans le menu qui s'ouvre.

Veillez à ce que les composants d'entrée soient tous liés correctement au composant ELT Map pour garantir que les inclusions, jointures et clauses fonctionnent.

Scénarios associés

Pour des scénarios associés, consultez les scénarios suivants du composant **tELTMysqlMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTSybaseOutput

Propriétés du tELTSybaseOutput

Les composants **tELTSybaseInput**, **tELTSybaseOutput**, et **tELTSybaseMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Sybase fin de générer des commandes Insert, y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille du composant	ELT/Map/Sybase	
Fonction	Effectue l'action sur la table telle que spécifiée et insère les données selon le schéma de sortie défini dans le Mapper ELT.	
Objectif	Exécute la requête Insert dans la base de données Sybase.	
Basic settings *Utilisez le composant tCreateTable pour remplacer cette option.*	*Action on data*	Sur les données de la table définie, vous pouvez effectuer les opérations suivantes : **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête. **Update** : Met à jour des lignes existantes. **Delete** : Supprime les entrées correspondantes au flux d'entrée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Le schéma défini est passé au Mapper ELT afin de l'insérer dans l'instruction SQL d'Insert. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Where clauses for (for UPDATE and DELETE only)	Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.
	Default Table Name	Saisissez le nom de la table par défaut.

	Default Schema Name	Saisissez le nom du schéma Sybase par défaut.
	Use different table name	Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ **Table name** qui apparaît.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTSybaseOutput** est étroitement liée à celle des composants **tELTSybaseInput** et **tELTSybaseMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour des scénarios associés, consultez les scénarios suivants du composant **tELTMysqlMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTTeradataInput

Propriétés de tELTTeradataInput

Les composants **tELTTeradataInput**, **tELTTeradataOutput**, et **tELTTeradataMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Teradata afin de générer des commandes Insert y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Teradata	
Fonction	Fournit le schéma de la table d'entrée à la commande SQL qui sera exécutée.	
Objectif	Permet d'ajouter autant de tables que nécessaires dans une commande Insert qui peut être complexe.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l'utiliser dans la commande SQL Insert. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Default Table Name	Saisissez le nom par défaut de la table, entre guillemets doubles.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	L'utilisation du composant **tELTTeradataInput** est étroitement liée à celle du **tELTTeradataMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

Scénarios associés

Pour des scénarios utilisant le composant **tELTTeradataInput**, consultez les scénarios suivants du **tELTTeradataMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTTeradataMap

Propriétés de tELTTeradataMap

Les composants **tELTTeradataInput**, **tELTTeradataOutput**, et **tELTTeradataMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Teradata afin de générer des commandes Insert y compris des clauses, et ces commandes devront être exécutées dans la table de sortie définie.

Famille de composant	ELT/Map/Teradata	
Fonction	Permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.	
Objectif	Utilise les tables fournies en entrée pour alimenter les paramètres nécessaires à l'exécution de la commande SQL souhaitée. La commande peut contenir des jointures internes comme externes entre différentes tables ou entre une table et ses alias.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.
	ELT Teradata Map editor	L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.
	Style link	Sélectionnez le type d'affichage des liens. **Auto** : par défaut, les liens entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbe. **Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne. **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d'optimiser les performances.
	Property type	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur de base de données.
	Database	Nom de la base de données.

	Username et *Password*	Données d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*Query band*	Cochez cette case pour utiliser la fonctionnalité Teradata Query Banding afin d'ajouter les métadonnées à la requête à traiter, comme le nom de l'utilisateur exécutant la requête. Cela peut vous permettre, par exemple, d'identifier l'origine de la requête. Une fois la case cochée, la table **Query Band parameters** s'affiche, dans laquelle vous pouvez saisir les informations des métadonnées à ajouter. Cette information prend la forme de paires clé/valeur, par exemple, *DpID* dans la colonne **Key** et *Finance* dans la colonne **Value**. Cette case permet de générer l'instruction SET QUERY_BAND FOR SESSION avec les paires clé/valeur déclarées dans la table **Query Band parameters**. Pour plus d'informations concernant cette instruction, consultez http://www.info.teradata.com/HTMLPubs/DB_TTU_14_00/index.html#page/SQL_Reference/B035_1144_111A/End_Logging-Syntax.027.143.html (en anglais). Cette case est indisponible lorsque vous avez coché la case **Using an existing connection**. Dans ce cas, si vous devez utiliser la fonctionnalité **Query band**, configurez-la dans l'onglet **Advanced settings** du composant de connexion Teradata à utiliser.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTTeradataMap** est étroitement liée à celle des composants **tELTTeradataInput** et **tELTTeradataOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	

Connecter les composants ELT

Pour plus d'informations concernant la connexion des composants ELT, consultez *Connecter les composants ELT*

Voir également le *Guide utilisateur du Studio Talend*.

Tables de jointure et de correspondance

Dans le Mapper ELT, vous pouvez sélectionner des colonnes des schémas d'entrée et les intégrer dans le schéma de sortie.

Pour plus d'informations concernant la mise en correspondance et les jointures, consultez *Tables de jointure et de correspondance*.

Ajouter des clauses WHERE et d'autres clauses

Vous pouvez également restreindre une requête Select à l'aide d'une clause WHERE et/ou à l'aide d'autres clauses telles que GROUP BY, ORDER BY, etc., en cliquant sur le bouton **Add filter row** situé en haut de la table de sortie dans l'éditeur de map.

Pour ajouter une restriction à l'aide d'une clause WHERE, cliquez sur le bouton **Add filter row** et sélectionnez **Add a WHERE clause** dans le menu qui s'ouvre.

Pour ajouter une restriction à l'aide de GROUP BY, ORDER BY, etc., cliquez sur le bouton **Add filter row** et sélectionnez **Add an other(GROUP...) clause** dans le menu qui s'ouvre.

Veillez à ce que les composants d'entrée soient tous liés correctement au composant ELT Map pour garantir que les inclusions, jointures et clauses fonctionnent.

Générer la requête SQL

La mise en correspondance d'éléments des schémas d'entrée avec le schéma de sortie crée automatiquement la requête Select correspondante.

La/les clause(s) définie(s) dans le mapper ELT est (sont) également intégrée(s) automatiquement dans la requête.

Scénario : Mapper des données à l'aide d'une sous-requête

Ce scénario décrit un Job mappant des données des deux tables d'entrée *PreferredSubject* et *CourseScore* vers la table de sortie *TotalScoreOfPreferredSubject* à l'aide d'une sous-requête.

La table *PreferredSubject* contient les données des matières préférées des étudiants. Pour reproduire ce scénario, vous pouvez charger des données dans la table à partir d'un fichier CSV comme suit. Pour plus d'informations concernant comment charger des données dans une table Teradata, consultez *Scénario : Charger des données dans une base de données Teradata*.

```
SeqID;StuName;Subject;Detail
1;Amanda;art;Amanda prefers art.
2;Ford;science;Ford prefers science.
3;Kate;art;Kate prefers art.
```

La table *CourseScore* contient les scores des étudiants dans leur matière favorite. Pour reproduire ce scénario, vous pouvez charger des données dans la table à partir d'un fichier CSV comme suit. Pour plus d'informations concernant comment charger des données dans une table Teradata, consultez *Scénario : Charger des données dans une base de données Teradata*.

```
SeqID;StuName;Subject;Course;Score;Detail
1;Amanda;science;math;85;science score
2;Amanda;science;physics;75;science score
3;Amanda;science;chemistry;80;science score
4;Amanda;art;chinese;85;art score
5;Amanda;art;history;95;art score
6;Amanda;art;geography;80;art score
7;Ford;science;math;95;science score
8;Ford;science;physics;85;science score
9;Ford;science;chemistry;80;science score
10;Ford;art;chinese;75;art score
11;Ford;art;history;80;art score
12;Ford;art;geography;85;art score
13;Kate;science;math;65;science score
14;Kate;science;physics;75;science score
15;Kate;science;chemistry;80;science score
16;Kate;art;chinese;85;art score
17;Kate;art;history;80;art score
18;Kate;art;geography;95;art score
```

Avant exécution du Job, il n'y a aucune donnée dans la table de sortie *TotalScoreOfPreferredSubject* :

```
SeqID;StuName;PreferredSubject;TotalScore
```

Déposer et renommer les composants

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant à partir de la **Palette** : deux **tELTTeradataInput**, deux **tELTTeradataMap** et un **tELTTeradataOutput**.

2. Renommez les **tELTTeradataInput** en *PreferredSubject* et *CourseScore* respectivement, les deux **tELTTeradataMap** en *ELTSubqueryMap* et *ELTMap* et le **tELTTeradataOutput** en *TotalScoreOfPreferredSubject*.

Configurer les composants d'entrée

1. Double-cliquez sur le composant *PreferredSubject* pour ouvrir sa vue **Basic settings**.

2. Dans le champ **Default Table Name**, saisissez un nom pour la table d'entrée. Dans cet exemple, nommez-la *PreferredSubject*.

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir le schéma de la table d'entrée dans l'éditeur du schéma.

Cliquez quatre fois sur le bouton **[+]** pour ajouter quatre colonnes : *SeqID* dont le **DB Type** est **INTEGER**, *StuName*, *Subject* et *Detail* dont le **DB Type** est **VARCHAR**.

Cliquez sur **OK** pour valider ces modifications et fermer l'éditeur du schéma.

4. Connectez le *PreferredSubject* au *ELTMap* à l'aide d'un lien **Link** > **PreferredSubject (Table)**.

5. Double-cliquez sur le *CourseScore* pour ouvrir sa vue **Basic settings**.

6. Dans le champ **Default Table Name**, saisissez le nom de la table d'entrée. Dans cet exemple, saisissez *CourseScore*.

7. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir le schéma de la table d'entrée dans l'éditeur du schéma.

Cliquez six fois sur le bouton **[+]** pour ajouter six colonnes, *SeqID* et *Score* dont le **DB Type** est **INTEGER**, *StuName*, *Subject*, *Course* et *Detail* dont le **DB Type** est **VARCHAR**.

Cliquez sur **OK** pour valider ces modifications et fermer l'éditeur du schéma.

8. Reliez le *CourseScore* au *ELTSubqueryMap* à l'aide d'un lien **Link** > **CourseScore (Table)**.

Configurer le composant de sortie

1. Double-cliquez sur le composant *TotalScoreOfPreferredSubject* pour ouvrir sa vue **Basic settings**.

2. Dans le champ **Default Table Name**, saisissez un nom pour la table de sortie. Dans cet exemple, saisissez *TotalScoreOfPreferredSubject*.

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin de définir le schéma de la table de sortie dans l'éditeur du schéma.

Cliquez quatre fois sur le bouton **[+]** pour ajouter quatre colonnes, *SeqID* et *TotalScore* dont le **DB Type** est **INTEGER**, *StuName* et *PreferredSubject* dont le **DB Type** est **VARCHAR**.

Cliquez sur **OK** pour valider ces modifications et fermer l'éditeur du schéma.

Configurer le mapping des données afin de générer une sous-requête

1. Double-cliquez sur le *ELTSubqueryMap* pour ouvrir sa vue **Basic settings**.

Vous n'avez pas besoin de spécifier les informations de connexion à la base de données Teradata dans le composant *ELTSubqueryMap*. Les informations de connexion sont spécifiées dans le composant *ELTMap*.

2. Cliquez sur le bouton **[...]** du champ **ELT Teradata Map Editor** pour ouvrir son éditeur.

3. Ajoutez la table d'entrée *CourseScore* en cliquant sur le bouton **[+]** dans le coin supérieur gauche de l'éditeur de mapping et sélectionnez le nom de la table dans la liste déroulante de la boîte de dialogue qui s'ouvre.

4. Ajoutez une table de sortie en cliquant sur le bouton **[+]** dans le coin supérieur droit de l'éditeur et en saisissant le nom de la table *TotalScore* dans le champ correspondant de la boîte de dialogue.

5. Cliquez sur les colonnes *StuName*, *Subject* et *Score* dans la table d'entrée et déposez-les dans la table de sortie.

6. Cliquez sur le bouton **Add filter row** dans le coin supérieur droit de la table de sortie et sélectionnez **Add an other(GROUP...) clause** dans le menu contextuel. Dans le champ **Additional other clauses (GROUP/ORDER BY...)**, saisissez la clause *GROUP BY CourseScore.StuName, CourseScore.Subject*.

   Ajoutez la fonction d'agrégation SUM pour la colonne *Score* de la table de sortie en modifiant l'expression de cette colonne en *SUM(CourseScore.Score)*.

7. Cliquez sur l'onglet **Generated SQL Select query for "table1" output** au bas de l'éditeur de mapping pour afficher l'instruction SQL correspondante générée.

   Cette requête SQL apparaît comme sous-requête SQL générée par le composant *ELTMap*.

8. Cliquez sur **OK** pour valider les modifications et fermer l'éditeur de mapping.

9. Reliez le *ELTSubqueryMap* au *ELTMap* à l'aide d'un lien **Link** > **TotalScore (table1)**. Notez que le lien est automatiquement renommé en **TotalScore (Table_ref)** car la table de sortie *TotalScore* est une table de référence.

Mapper les schémas d'entrée et de sortie

1. Cliquez-droit sur le composant *ELTMap*, sélectionnez **Link** > ***New Output* (Table)** dans le menu contextuel et cliquez sur *TotalScoreOfPreferredSubject*. Dans la boîte de dialogue qui s'ouvre, cliquez sur **Yes** pour récupérer le schéma du composant cible.

2. Cliquez sur le *ELTMap* pour ouvrir sa vue **Basic settings**.

Dans les champs **Host**, **Database**, **Username** et **Password**, saisissez vos informations de connexion à la base de données Teradata.

3. Cliquez sur le bouton **[...]** à côté du champ **ELT Teradata Map Editor** pour ouvrir son éditeur.

4. Ajoutez la table d'entrée *PreferredSubject* en cliquant sur le bouton **[+]** dans le coin supérieur gauche de l'éditeur et sélectionnez le nom de la table souhaitée dans la liste de la boîte de dialogue.

   Répétez l'opération pour ajouter une autre table d'entrée, *TotalScore*.

5. Cliquez sur la colonne *StuName* de la table d'entrée *PreferredSubject* et déposez-la dans la colonne correspondante de la table d'entrée *TotalScore*. Cochez la case **Explicit join** de la colonne *StuName* dans la table *TotalScore*.

   Répétez l'opération avec la colonne *Subject*.

6. Cliquez sur la colonne *SeqID* de la table d'entrée *PreferredSubject* et déposez-la dans la colonne correspondante de la table de sortie.

   Répétez l'opération avec les colonnes *StuName* et *Subject* de la table d'entrée *PreferredSubject* et la colonne *Score* de la table d'entrée *TotalScore*.

7. Cliquez sur l'onglet **Generated SQL Select query for "table2" output** au bas de l'éditeur de mapping pour afficher l'instruction SQL correspondante générée.

   Schema editor | Expression editor | Generated SQL Select query for "table2" output

```
"SELECT
PreferredSubject.SeqID , PreferredSubject.StuName , PreferredSubject.Subject , TotalScore.Score
FROM
 PreferredSubject INNER JOIN  (
  SELECT
  CourseScore.StuName  AS StuName, CourseScore.Subject  AS Subject, SUM(CourseScore.Score) AS Score
  FROM
   CourseScore
  GROUP BY CourseScore.StuName, CourseScore.Subject

 ) TotalScore ON(  TotalScore.StuName EQ PreferredSubject.StuName  AND  TotalScore.Subject EQ PreferredSubject.Subject  )"
```

   La requête SQL générée dans le composant *ELTSubqueryMap* apparaît comme sous-requête dans la requête QSL générée par ce composant. L'alias est automatiquement ajouté pour les colonnes sélectionnées dans la sous-requête

8. Cliquez sur **OK** pour valider les modifications et fermer l'éditeur de mapping.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.

2. Appuyez sur **F6** pour l'exécuter.

```
Starting job DOCT3563_tELTTeradataMap at 16:29 17/10/2014.

[statistics] connecting to socket on port 3751
[statistics] connected
Inserting with :
INSERT INTO TotalScoreOfPreferredSubject
(SeqID,StuName,PreferredSubject,TotalScore) SELECT
PreferredSubject.SeqID , PreferredSubject.StuName ,
PreferredSubject.Subject , TotalScore.Score  FROM  PreferredSubject
INNER JOIN  (   SELECT   CourseScore.StuName  AS StuName,
CourseScore.Subject  AS Subject, SUM(CourseScore.Score) AS Score   FROM
CourseScore   GROUP BY CourseScore.StuName, CourseScore.Subject     )
TotalScore ON(  TotalScore.StuName EQ PreferredSubject.StuName  AND
TotalScore.Subject EQ PreferredSubject.Subject  )
--> 0 rows inserted.
[statistics] disconnected
Job DOCT3563_tELTTeradataMap ended at 16:29 17/10/2014. [exit code=0]
```

	SeqID	StuName	PreferredSubject	TotalScore
1	1	Amanda	art	260
2	2	Ford	science	260
3	3	Kate	art	260

L'instruction sélectionnée est générée et les données de mapping sont écrites dans la table de sortie.

Scénarios associés

Pour des scénarios associés, consultez les scénarios suivants du composant **tELTMysqlMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tELTTeradataOutput

Propriétés de tELTTeradataOutput

Les composants **tELTTeradataInput**, **tELTTeradataOutput**, et **tELTTeradataMap** sont très proches dans leurs conditions de fonctionnement. En effet, ces trois composants sont conçus pour manipuler des schémas de données d'une base Teradata afin de générer des commandes Insert y compris des clauses, qui devront être exécutées dans la table de sortie définie.

Famille du composant	ELT/Map/Teradata	
Fonction	Effectue l'action sur la table telle que spécifiée et insère les données selon le schéma de sortie défini dans le Mapper ELT.	
Objectif	Exécute la requête Insert ou Update dans la base de données Teradata.	
Basic settings	*Action on data*	Sur les données de la table définie, vous pouvez effectuer les opérations suivantes : **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête. **Update** : Met à jour des lignes existantes. **Delete** : Supprime les entrées correspondantes au flux d'entrée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Le schéma défini est passé au Mapper ELT afin de l'insérer dans l'instruction SQL d'Insert. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Where clauses for (for UPDATE and DELETE only)	Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.
	Default Table Name	Saisissez le nom par défaut de la table, entre guillemets doubles.

	Use different table name	Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ **Table name** qui apparaît.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	L'utilisation du composant **tELTTeradataOutput** est étroitement liée à celle des composants **tELTTeradataInput** et **tELTTeradataMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour des scénarios associés, consultez les scénarios suivants du composant **tELTMysqlMap** :

- *Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre*.
- *Scénario 2 : ELT utilisant une table Alias*.

tFirebirdConnection

Le composant **tFirebirdConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tFirebirdConnection*.

tGreenplumConnection

Le composant **tGreenplumConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tGreenplumConnection*.

tHiveConnection

Le composant **tHiveConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tHiveConnection*.

tIngresConnection

Le composant **tIngresConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tIngresConnection*.

tInterbaseConnection

Le composant **tInterbaseConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tInterbaseConnection*.

tJDBCConnection

Le composant **tJDBCConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tJDBCConnection*.

tMSSqlConnection

Le composant **tMSSqlConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tMSSqlConnection*.

tMysqlConnection

Le composant **tMysqlConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tMysqlConnection*.

tNetezzaConnection

Le composant **tNetezzaConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tNetezzaConnection*.

tOracleConnection

Le composant **tOracleConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tOracleConnection*.

tParAccelConnection

Le composant **tParAccelConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tParAccelConnection*.

tPostgresPlusConnection

Le composant **tPostgresPlusConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tPostgresPlusConnection*.

tPostgresqlConnection

Le composant **tPostgresqlConnection** appartient à deux familles : Databases et ELT. Pour plus d'informations, consultez *tPostgresqlConnection*.

tSAPHanaConnection

Le composant **tSAPHanaConnection** appartient à deux familles : Databases et ELT. Pour plus d'informations, consultez *tSAPHanaConnection*.

tSQLiteConnection

Le composant **tSQLiteConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tSQLiteConnection*.

tSQLTemplate

Propriétés du tSQLTemplate

Famille de composant	ELT/SQLTemplate	
Fonction	Le composant **tSQLTemplate** fournit un ensemble de modèles d'instructions SQL pour un certain nombre de SGBD, afin de faciliter les actions les plus communes sur une base de données. De plus, il est possible de personnaliser les modèles d'instructions SQL selon vos besoins.	
Objectif	Ce composant permet aux utilisateurs d'exécuter simplement des actions sur une base de données ou des modèles d'instructions SQL personnalisés, par exemple pour supprimer ou créer une table. Ces modèles personnalisés sont accessibles via la vue **SQL Template**.	
Basic settings	*Database Type*	Sélectionnez dans la liste le type de base de données à laquelle vous connecter.
	Component List	Sélectionnez dans la liste le composant de connexion à la base de données à utiliser, si plus d'une connexion est utilisée dans le Job.
	Database name	Saisissez le nom de la base de données.
	Table name	Saisissez le nom de la table sur laquelle exécuter les modèles d'instructions SQL.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log aussi bien au niveau du Job qu'au niveau des composants.
SQL Template	*SQL Template List*	**Pour ajouter un modèle système SQL par défaut :** Cliquez sur le bouton **Add** pour ajouter le(s) modèle(s) système par défaut dans la table **SQL Template List**. Cliquez dans le champ **SQL template** puis cliquez sur la flèche pour afficher la liste des modèles SQL système. Sélectionnez le modèle de votre choix, fourni par **Talend**.

		Note : Vous pouvez créer vos propres modèles SQL et les ajouter à la table **SQL Template List**. **Pour créer un modèle SQL personnalisé :** - Sélectionnez un modèle système dans la table **SQL Template list** et cliquez sur son code dans le champ **Code**. Le système vous propose de créer un nouveau modèle. - Cliquez sur **Yes** pour ouvrir l'assistant des modèles SQL. - Configurez les paramètres de votre modèle SQL puis cliquez sur **Finish** pour fermer l'assistant. Un éditeur de modèle SQL s'ouvre, dans lequel vous pouvez saisir le code du modèle. - Cliquez sur le bouton **Add** pour ajouter le nouveau modèle à la table **SQL Template list**. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de début. Il doit être utilisé avec d'autres composants de bases de données, notamment des composants de connexion et de commit.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD*.

tSQLTemplateAggregate

Propriétés du tSQLTemplateAggregate

Famille de composant	ELT/SQLTemplate	
Fonction	Le composant **tSQLTemplateAggregate** collecte les données d'une ou plusieurs colonnes et gère ces données en un seul bloc. Ce composant exécute en temps réel des transformations de données dans le SGBD lui-même.	
Objectif	Il permet d'établir des métriques basées sur des valeurs ou des calculs.	
Basic settings	*Database Type*	Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.
	Component List	Sélectionnez le composant de connexion à la base de données dans la liste si vous utilisez plus d'une connexion dans le Job en cours.
	Database name	Nom de la base de données.
	Source table name	Nom de la table contenant les données dont vous souhaitez collecter les données.
	Target table name	Nom de la table dans laquelle vous souhaitez écrire les données collectées et transformées.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Operations	Sélectionnez le type d'opération et la valeur à utiliser pour le calcul et le champ de sortie.
		Output Column : Sélectionnez le champ de destination dans la liste.
		Function : Sélectionnez une des opérations suivantes à effectuer sur les données : **count :** calcule le nombre de lignes, **min :** sélectionne la plus petite valeur,

		max : sélectionne la plus grande valeur, **avg :** calcule la moyenne, **sum :** calcule la somme, **count (distinct) :** compte le nombre de lignes sans les doublons.
		Input column position : Sélectionnez la colonne d'entrée à partir de laquelle vous souhaitez collecter les données à agréger.
	Group by	Définit les ensembles d'agrégation, dont les valeurs sont utilisées pour les calculs.
		Output Column : Sélectionnez le libellé de colonne dans la liste fournie, basée sur la structure de schéma que vous avez définie. Vous pouvez ajouter autant de colonnes de sortie que vous le souhaitez afin d'affiner les agrégations.
		Input Column position : Faites la correspondance entre les libellés des colonnes d'entrée avec ceux des colonnes de sortie, dans le cas où vous souhaitez que les libellés du schéma de sortie soient différents du schéma d'entrée.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
SQL Template	*SQLTemplate List*	Cliquez sur le bouton **Add** pour ajouter un ou plusieurs modèle SQL du système à la liste **SQLTemplateList**. Cliquez dans la cellule de la colonne **SQLTemplate List** que vous venez d'ajouter et sélectionnez le modèle SQL du système adéquat dans la liste de modèles fournis par **Talend**. **Note :** Vous pouvez créer vos propres modèles SQL et les ajouter à la liste **SQLTemplate List**. Pour créer un nouveau modèle SQL : 1- Sélectionnez un modèle déjà existant dans la liste **SQLTemplate list** et cliquez dans son code dans le champ de code. Une boîte de dialogue s'ouvrira vous demandant de créer un nouveau modèle. 2- Cliquez sur **Yes** pour ouvrir la boîte de dialogue **[New SQLTemplate]**. 3- Renseignez les informations de votre modèle SQL et cliquez sur **Finish** pour fermer la boîte de dialogue. Un éditeur de modèle SQL apparaît dans lequel vous pouvez saisir votre code. 4- Cliquez sur le bouton **Add** pour ajouter le nouveau modèle dans la liste **SQLTemplate list**. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant intermédiaire. Vous pouvez l'utiliser avec d'autres composants de base de données, notamment les composants de connexion et de commit.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD

Le scénario suivant crée un Job qui ouvre une connexion à une base de données Mysql et :

- crée un schéma à partir de la table d'une base de données, les lignes de ce schéma correspondent aux noms des colonnes spécifiées dans le filtre,
- filtre les colonnes de cette même table afin de n'obtenir que les données correspondant aux colonnes filtrées,
- collecte les données de la colonne filtrée, regroupées en fonction d'une valeur spécifique et écrit les données agrégées dans la table de cible.

Pour filtrer et agréger les colonnes de la table de votre base de données :

- Cliquez et déposez les composants suivants dans l'espace de modélisation : **tELTMysqlconnection**, **tSQLTemplateFilterColumns**, **tSQLTemplateFilterRows**, **tSQLTemplateAggregate**, **tSQLTemplateCommit** et **tSQLTemplateRollback**.
- Connectez les cinq composants à l'aide de liens de type **OnComponentOk**.
- Connectez les composants **tSQLTemplateAggregate** et **tSQLTemplateRollback** à l'aide d'un lien de type **OnComponentError**.

- Dans l'espace de modélisation, sélectionnez le **tMysqlConnection** et cliquez sur la vue **Component** pour paramétrer ses propriétés.
- Dans l'onglet **Basic settings**, renseignez manuellement les informations de connexion ou sélectionnez-les à partir des variables de contexte, via la raccourci **Ctrl+Espace** cliquez sur le champ correspondant si les vous avez stocké sous le nœud **DB connection** du répertoire **Metadata** du **Repository**.

Pour plus d'informations sur les métadonnées, consultez le *Guide utilisateur du Studio Talend*.

- Dans l'espace de modélisation, sélectionnez le composant **tSQLTemplateFilterColumns** et cliquez sur la vue **Component** pour paramétrer ses propriétés.

- Dans la liste déroulante **Database type**, sélectionnez la base de données adéquate.
- Dans la liste déroulante **Component list**, sélectionnez le composant de connexion à la base de données si plusieurs composants de connexion sont utilisés dans votre Job.
- Dans le champ **Database name**, saisissez le nom de la base de données à laquelle vous vous êtes connecté.
- Dans le champ **Source table name**, saisissez le nom de votre table source contenant les données que vous souhaitez traiter et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir la structure des données de votre table source.
- Dans le champ **Target table name**, saisissez le nom de la table cible contenant vos nouvelles données traitées et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir la structure de la table cible.

Lorsque vous définissez la structure des données de la table source, le nom des colonnes apparaît automatiquement dans la colonne **Column** du tableau **Column filters**.

Dans ce scénario, la table source contient les cinq colonnes : *id*, *First_Name*, *Last_Name*, *Address* et *id_State*.

- Dans le tableau **Column filters**, filtrez les colonnes en cochant la case des colonnes que vous souhaitez écrire dans votre table cible.

Dans ce scénario, le composant **tSQLTemplateFilterColumns** ne garde que les colonnes *id*, *First_Name* et *id_State* de la table source.

Dans la vue **Component**, vous pouvez cliquer sur l'onglet **SQL Template** et ajouter des modèles SQL du système ou créer vos propres modèles SQL et les utilisez dans votre Job afin d'effectuer les opérations que vous souhaitez. Pour plus d'informations, consultez *Propriétés du tSQLTemplateFilterColumns*.

- Dans l'espace de modélisation, sélectionnez le composant **tSQLTemplateFilterRows** et cliquez sur l'onglet **Component** pour paramétrer ses propriétés.

- Dans la liste déroulante **Database type**, sélectionnez la base de données adéquate.
- Dans la liste déroulante **Component list**, sélectionnez le composant de connexion à la base de données si plusieurs composants de connexion sont utilisés dans votre Job.
- Dans le champ **Database name**, saisissez le nom de la base de données à laquelle vous vous êtes connecté.
- Dans le champ **Source table name**, saisissez le nom de votre table source contenant les données que vous souhaitez traiter et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir la structure des données de votre table source.
- Dans le champ **Target table name**, saisissez le nom de la table cible contenant vos nouvelles données traitées et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir la structure de la table cible.

Dans ce scénario, la table source et la table cible contiennent les trois colonnes *filtrées id*, *First_Name* et *id_State*.

- Dans le champ **Where condition**, saisissez la commande WHERE vous permettant de n'extraire que les champs correspondant à vos critères.

Dans ce scénario, le composant **tSQLTemplateFilterRows** filtre la colonne *First_Name* de la table source pour n'obtenir que les prénoms commencent par la lettre "a".

- Dans l'espace de modélisation, sélectionnez le composant **tSQLTemplateAggregate** et cliquez sur la vue **Component** pour paramétrer ses propriétés.
- Dans la liste déroulante **Database type**, sélectionnez la base de données adéquate.
- Dans la liste **Component list**, sélectionnez le composant de connexion à la base de données si plusieurs composants de connexion sont utilisés dans votre Job.
- Dans le champ **Database name**, saisissez le nom de la base de données à laquelle vous vous êtes connecté.
- Dans le champ **Source table name**, saisissez le nom de votre table source contenant les données que vous souhaitez traiter et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir la structure des données de votre table source.
- Dans le champ **Target table name**, saisissez le nom de la table cible contenant vos nouvelles données traitées et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir la structure de la table cible.

Le schéma de la table source est composé de trois colonnes : *id*, *First_Name* et *id_State*. Le schéma de la table cible est composé de deux colonnes : *customers_status* et *customers_number*. Dans ce scénario, vous souhaitez regrouper les clients par statut marital et compter le nombre de client pour chaque groupe marital. Pour cela, renseignez les tableaux **Operations** et **Group by**.

- Dans le tableau **Operations**, cliquez sur le bouton **[+]** pour ajouter une ou plusieurs lignes et cliquez dans la cellule de la colonne **Output column** et sélectionnez la colonne de sortie qui contiendra les données à compter.
- Cliquez dans la cellule de la colonne **Function** et sélectionnez l'opération à effectuer.
- Dans le tableau **Group by**, cliquez sur le bouton **[+]** pour ajouter une ou plusieurs lignes et cliquez dans la cellule de la colonne **Output column** pour sélectionner la colonne de sortie qui contiendra les données agrégées.
- Dans l'espace de modélisation, sélectionnez le composant **tSQLTemplateCommit** et cliquez sur la vue **Component** pour paramétrer ses propriétés.
- Dans la liste déroulante **Database type**, sélectionnez la base de données adéquate.
- Dans la liste déroulante **Component list**, sélectionnez le composant de connexion à la base de données si plusieurs composants de connexion sont utilisés dans votre Job.
- Effectuez la même procédure pour le composant **tSQLTemplateRollback**.
- Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

La table *aggregate_customers* composée de deux colonnes est créée dans la base de données. Elle regroupe les clients en fonction de leur statut marital et compte le nombre de client pour chaque statut marital.

tSQLTemplateCommit

Propriétés du tSQLTemplateCommit

Famille du composant	ELT/SQLTemplate	
Fonction	Le composant **tSQLTemplateCommit** valide les données traitées dans un Job à partir d'une base de données connectée.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.	
Basic settings	*Database Type*	Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.
	Component List	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant de connexion à la base de données que vous utilisez dans la liste.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
SQL Template	*SQLTemplate List*	Cliquez sur le bouton **Add** pour ajouter un ou plusieurs modèle SQL du système à la liste **SQLTemplateList**. Cliquez dans la cellule de la colonne **SQLTemplate List** que vous venez d'ajouter et sélectionnez le modèle SQL du système adéquat dans la liste de modèles fournis par **Talend**. **Note :** Vous pouvez créer vos propres modèles SQL et les ajouter à la liste **SQLTemplate List**. Pour créer un nouveau modèle SQL : 1- Sélectionnez un modèle déjà existant dans la liste **SQLTemplate list** et cliquez dans son code dans le champ de code. Une boîte de dialogue s'ouvrira vous demandant de créer un nouveau modèle. 2- Cliquez sur **Yes** pour ouvrir la boîte de dialogue **[New SQLTemplate]**. 3- Renseignez les informations de votre modèle SQL et cliquez sur **Finish** pour fermer la boîte de dialogue. Un éditeur de modèle SQL apparaît dans lequel vous pouvez saisir votre code. 4- Cliquez sur le bouton **Add** pour ajouter le nouveau modèle dans la liste **SQLTemplate list**. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé avec des composants ELT et notamment avec le composant **tSQLTemplateRollback,** ainsi qu'avec le composant ELT de connexion à la base de données que vous utilisez.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tSQLTemplateCommit**, consultez *Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD*.

tSQLTemplateFilterColumns

Propriétés du tSQLTemplateFilterColumns

Famille de composant	ELT/SQLTemplate	
Fonction	Le composant **tSQLTemplateFilterColumns** modifie le schéma de la table d'une base de données en filtrant les colonnes. Ce composant filtre en temps réel les données dans le SGBD lui-même.	
Objectif	Il permet d'homogénéiser des schémas en réorganisant ou en supprimant des colonnes ou en ajoutant de nouvelles colonnes.	
Basic settings	*Database Type*	Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.
	Component List	Sélectionnez le composant de connexion à la base de données si vous utilisez plus d'une connexion dans votre Job en cours.
	Database name	Nom de la base de données.
	Source table name	Nom de la table contenant les données à filtrer.
	Target table name	Nom de la table dans laquelle vous souhaitez écrire les données filtrées.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Column filters	Dans le tableau, cochez la case **Filter** pour filtrer toutes vos colonnes, ou, pour en filtrer moins, cliquez sur la ou les case(s) en face du nom de la ou des colonne(s).
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
SQL Template	*SQLTemplate List*	Cliquez sur le bouton **Add** pour ajouter un ou plusieurs modèle SQL du système à la liste **SQLTemplateList**. Cliquez dans la cellule de la colonne **SQLTemplate List** que vous venez d'ajouter et sélectionnez le modèle SQL du système adéquat dans la liste de modèles fournis par **Talend**.

		Note : Vous pouvez créer vos propres modèles SQL et les ajouter à la liste **SQLTemplate List**. Pour créer un nouveau modèle SQL : 1- Sélectionnez un modèle déjà existant dans la liste **SQLTemplate list** et cliquez dans son code dans le champ de code. Une boîte de dialogue s'ouvrira vous demandant de créer un nouveau modèle. 2- Cliquez sur **Yes** pour ouvrir la boîte de dialogue **[New SQLTemplate]**. 3- Renseignez les informations de votre modèle SQL et cliquez sur **Finish** pour fermer la boîte de dialogue. Un éditeur de modèle SQL apparaît dans lequel vous pouvez saisir votre code. 4- Cliquez sur le bouton **Add** pour ajouter le nouveau modèle dans la liste **SQLTemplate list**. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire. Vous pouvez l'utiliser avec d'autres composants de base de données, notamment les composants de connexion et de commit.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD*.

tSQLTemplateFilterRows

Propriétés du tSQLTemplateFilterRows

Famille de composant	ELT/SQLTemplate	
Fonction	Le composant **tSQLTemplateFilterRows** vous permet de filtrer les lignes d'une table. Ce composant filtre en temps réel les données dans le SGBD lui-même.	
Objectif	Il permet de mettre en place un filtrage de lignes sur n'importe quelle donnée source grâce à une commande WHERE.	
Basic settings	*Database Type*	Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.
	Component List	Sélectionnez le composant de connexion à la base de données si vous utilisez plus d'une connexion dans le Job en cours.
	Database name	Nom de la base de données.
	Source table name	Nom de la table contenant les données à filtrer.
	Target table name	Nom de la table dans laquelle vous souhaitez écrire les données filtrées.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Where condition	Utilisez une commande WHERE pour définir un critère à partir duquel vous souhaitez filtrer vos données. Vous pouvez utiliser la commande WHERE pour sélectionner des lignes spécifiques de votre table en fonction de certains critères ou certaines conditions.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log aussi bien au niveau du Job qu'au niveau du composant.
SQL Template	*SQLTemplate List*	Cliquez sur le bouton **Add** pour ajouter un ou plusieurs modèle SQL du système à la liste **SQLTemplateList**. Cliquez dans la

		cellule de la colonne **SQLTemplate List** que vous venez d'ajouter et sélectionnez le modèle SQL du système adéquat dans la liste de modèles fournis par **Talend**. **Note :** Vous pouvez créer vos propres modèles SQL et les ajouter à la liste **SQLTemplate List**. Pour créer un nouveau modèle SQL : 1- Sélectionnez un modèle déjà existant dans la liste **SQLTemplate list** et cliquez dans son code dans le champ de code. Une boîte de dialogue s'ouvrira vous demandant de créer un nouveau modèle. 2- Cliquez sur **Yes** pour ouvrir la boîte de dialogue **[New SQLTemplate]**. 3- Renseignez les informations de votre modèle SQL et cliquez sur **Finish** pour fermer la boîte de dialogue. Un éditeur de modèle SQL apparaît dans lequel vous pouvez saisir votre code. 4- Cliquez sur le bouton **Add** pour ajouter le nouveau modèle dans la liste **SQLTemplate list**. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire. Vous pouvez l'utiliser avec d'autres composants de base de données, notamment les composants de connexion et de commit.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD*.

tSQLTemplateMerge

Propriétés du tSQLTemplateMerge

Famille de composant	ELT/SQLTemplate	
Fonction	Le composant **tSQLTemplateMerge** crée une instruction SQL MERGE sur les données dans une table de la base de données.	
Objectif	Ce composant est utilisé pour effectuer des opérations de MERGE sur les données (insertion, mise à jour ou suppression) directement dans une table du SGBD. Pour cela, il crée et exécute une instruction MERGE.	
Basic settings	*Database Type*	Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.
	Component list	Sélectionnez le composant ELT de connexion le plus pertinent dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Source table name	Nom de la table contenant les données sur lesquelles vous souhaitez effectuer les opérations d'insertion ou de mise à jour dans la table cible.
	Target table name	Nom de la table dans laquelle vous souhaitez effectuer les opérations d'insertion ou de mise à jour.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Merge ON	Spécifiez les colonnes source et cible que vous souhaitez utiliser comme clés primaires.
	Use UPDATE (WHEN MATCHED)	Cochez cette case pour mettre à jour les enregistrements existants. Une fois cette case cochée, la table **UPDATE Columns** apparaît, vous permettant de définir les colonnes dans lesquelles les enregistrements seront mis à jour.
	Specify additional output columns	Cochez cette case pour mettre à jour les enregistrements dans des colonnes autres que celles listées dans la table **UPDATE Columns**.

		Une fois cette case cochée, la table **Additional UPDATE Columns** apparaît, vous permettant de définir des colonnes supplémentaires.
	Specify UPDATE WHERE clause	Cochez cette case et saisissez une clause de type WHERE dans le champ **WHERE clause** afin de filtrer les données lors de l'opération de mise à jour. Il est probable que cette option ne fonctionne pas avec certaines versions des bases de données, notamment Oracle 9i.
	Use INSERT (WHEN MATCHED)	Cochez cette case pour insérer de nouveaux enregistrements. Une fois cette case cochée, la table **INSERT Columns** apparaît, vous permettant de définir des colonnes à impliquer dans l'opération d'insertion.
	Specify additional output columns	Cochez cette case pour insérer des enregistrements dans les colonnes des colonnes autres que celles listées dans la table **INSERT Columns**. Une fois cette case cochée, la table **Additional INSERT Columns** apparaît, vous permettant de définir des colonnes supplémentaires.
	Specify INSERT WHERE clause	Cochez cette case et saisissez une clause de type WHERE dans le champ **WHERE clause** fin de filtrer les données lors de l'opération d'insertion. Il est probable que cette option ne fonctionne pas avec certaines versions des bases de données, notamment Oracle 9i.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log aussi bien au niveau du Job qu'au niveau du composant.
SQL Template	*SQL Template List*	Cliquez sur le bouton **Add** pour ajouter un ou plusieurs modèle SQL du système à la liste **SQLTemplateList**. Cliquez dans la cellule de la colonne **SQLTemplate List** que vous venez d'ajouter et sélectionnez le modèle SQL du système adéquat dans la liste de modèles fournis par **Talend**. **Note :** Vous pouvez créer vos propres modèles SQL et les ajouter à la liste **SQLTemplate List**. Pour créer un nouveau modèle SQL : 1- Sélectionnez un modèle déjà existant dans la liste **SQLTemplate list** et cliquez dans son code dans le champ de code. Une boîte de dialogue s'ouvrira vous demandant de créer un nouveau modèle. 2- Cliquez sur **Yes** pour ouvrir la boîte de dialogue **[New SQLTemplate]**. 3- Renseignez les informations de votre modèle SQL et cliquez sur **Finish** pour fermer la boîte de dialogue. Un éditeur de modèle SQL apparaît dans lequel vous pouvez saisir votre code. 4- Cliquez sur le bouton **Add** pour ajouter le nouveau modèle dans la liste **SQLTemplate list**. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_MERGED** : nombre de lignes fusionnées. Cette variable est une variable **After** et retourne un nombre entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant intermédiaire. Vous pouvez l'utiliser avec d'autres composants de base de données, notamment les composants de connexion et de commit.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Effectuer des opérations de MERGE directement dans le SGBD

Ce scénario décrit un Job simple permettant d'ouvrir une connexion à une base de données MySQL, d'effectuer des opérations d'insertion ou de mise à jour sur les données d'une table source vers une table cible en fonction de l'identifiant de l'utilisateur et d'afficher le contenu de la table cible avant et après les opérations de MERGE. Une clause de type WHERE est utilisée pour filtrer les données lors de ces opérations.

- A partir de la **Palette,** déposez les composants suivants dans l'espace de modélisation graphique : un **tMysqlConnection**, un **tSQLTemplateMerge**, deux **tMysqlInput** et deux **tLogRow**.
- Connectez le composant **tMysqlConnection** au premier **tMysqlInput** à l'aide d'un lien **Trigger** > **OnSubjobOk**.
- Connectez le premier **tMysqlInput** au premier **tLogRow** à l'aide d'un lien **Row** > **Main**. Cette ligne affichera le contenu initial de la table cible dans la console de la vue **Run**.

- Connectez le premier **tMysqlInput** au composant **tSQLTemplateMerge**, et le **tSQLTemplateMerge** au second composant **tMysqlInput** à l'aide de connexions **Trigger** > **OnSubjobOk**.

- Connectez le second **tMysqlInput** au second **tLogRow** à l'aide d'un lien **Row** > **Main**. Cette ligne affichera le résultat des opérations de MERGE dans la console de la vue **Run**.

- Double-cliquez sur le composant **tMysqlConnection** pour afficher l'onglet **Basic settings** de sa vue **Component**.

- Paramétrez manuellement les détails de la connexion à la base de données ou sélectionnez **Repository** dans la liste **Property Type** et sélectionnez votre connexion à une base de données si celle-ci a déjà été définie et stockée dans la zone **Metadata** de la vue **Repository**.

  Pour plus d'informations concernant les Métadonnées, consultez le *Guide utilisateur du Studio Talend*.

- Double-cliquez sur le premier composant **tMysqlInput** pour afficher l'onglet **Basic settings** de sa vue **Component**.

- Cochez la case **Use an existing connection**. Si vous souhaitez utiliser plus d'un composant de connexion dans votre Job, sélectionnez le composant que vous souhaitez utiliser à partir de la liste **Component List**.

- Cliquez sur le bouton **[...]** à côté de **Edit schema** et définissez la structure des données de la table cible, ou sélectionnez **Repository** à partir de la liste **Schema** puis sélectionnez la table cible si le schéma a déjà été défini et stocké dans la zone **Metadata** de la vue **Repository**.

  Pour ce scénario, choisissez **Built-in**.

- Définissez les colonnes comme indiqué ci-dessus, puis cliquez sur **OK** pour propager la structure du schéma au composant de sortie et fermer la boîte de dialogue.
- Renseignez le champ **Table Name** avec le nom de la table cible, soit *customer_info_merge* pour ce scénario.
- Cliquez sur le bouton **Guess Query**, ou saisissez `"SELECT * FROM customer_info_merge"` dans la zone **Query** afin de récupérer toutes les colonnes des tables.
- Définissez les propriétés du second composant **tMysqlInput** en utilisant exactement les mêmes paramètres que dans le premier composant **tMysqlInput**.
- Dans l'onglet **Basic settings** de la vue **Component** de chaque composant **tLogRow**, sélectionnez l'option **Table** dans la zone **Mode** afin d'afficher le résultat sous forme de tableau dans la console de la vue **Run**.

- Double-cliquez sur le composant **tSQLTemplateMerge** pour afficher l'onglet **Basic settings** de sa vue **Component**.

- Saisissez les noms de la table source et de la table cible dans les champs correspondants.

  Pour ce scénario, la table source, *new_customer_info*, contient huit enregistrements ; la table cible, *customer_info_merge*, contient cinq enregistrements et les deux tables ont la même structure.

  Il est possible que la table source et la table cible aient des schémas différents. Dans ce cas, cependant, assurez-vous que la colonne source et la colonne cible spécifiées dans chaque ligne des tables **Merge ON**, **UPDATE Columns** et **INSERT Columns** aient un type de données identique et que la longueur de la colonne cible permette l'insertion de données à partir de la colonne source correspondante.

- Définissez le schéma source manuellement, ou sélectionnez **Repository** à partir de la liste **Schema** et sélectionnez la table souhaitée si le schéma a déjà été défini et stocké dans la zone **Metadata** de la vue **Repository**.

  Pour ce scénario, des schémas en mode **Built-in** sont utilisés.

- Définissez les colonnes comme indiqué ci-dessus puis cliquez sur **OK** pour fermer la boîte de dialogue. Faites de même pour le schéma cible.

- Cliquez sur le bouton **[+]** sous la table **Merge ON** pour ajouter une ligne et sélectionnez la colonne *ID* comme clé primaire.

- Cochez la case **Use UPDATE (WHEN MATCHED)** pour mettre à jour les données existantes lors des opérations de MERGE, définissez les colonnes à mettre à jour en cliquant sur le bouton **[+]** et sélectionnez les colonnes souhaitées.

  Le but de ce scénario est de mettre à jour toutes les colonnes selon l'identifiant (ID) des utilisateurs. Pour cela, sélectionnez toutes les colonnes sauf la colonne *ID*.

  Les colonnes définies comme clés primaires ne peuvent pas et ne doivent pas être mises à jour.

- Cochez la case **Specify UPDATE WHERE clause** et saisissez `customer_info_merge.ID >= 4` entre guillemets doubles, dans le champ **WHERE clause** afin que seuls les enregistrements existants ayant un ID égal ou supérieur à 4 soient mis à jour.

- Cochez la case **Use INSERT** puis définissez dans la table **INSERT Columns** les colonnes où les données seront prises et insérées.

  Pour ce scénario, tous les enregistrements qui n'existent pas dans la table cible sont insérés.

- Sélectionnez l'onglet **SQL Template** pour afficher et ajouter les modèles SQL à utiliser.

  Par défaut, le composant **SQLTemplateMerge** utilise deux modèles SQL système : **MergeUpdate** et **MergeInsert**.

Dans l'onglet **SQL Template**, vous pouvez ajouter des modèles SQL système ou créer vos propres modèles et les utiliser dans votre Job pour effectuer des opérations codées. Pour plus d'informations, consultez *Propriétés du tSQLTemplateFilterColumns*.

- Cliquez sur le bouton **Add** pour ajouter une ligne puis sélectionnez **Commit** dans la liste des modèles afin de commiter les résultats des opérations de MERGE dans votre base de données.

  Autrement, vous pouvez connecter le composant **tSQLTemplateMerge** à un **tSQLTemplateCommit** ou un **tMysqlCommit** à l'aide d'un lien **Trigger** > **OnSubjobOk** afin de commiter les résultats des opérations dans votre base de données.

- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

  La console de la vue **Run** affiche à la fois le contenu d'origine de la table cible ainsi que les résultats des opérations de MERGE. Dans la table cible, les enregistrements No. 4 et No. 5 contiennent les informations mises à jour, tandis que les enregistrements No .6, No. 7 et No. 8 contiennent les informations insérées dans la table.

```
Starting job MergeData at 10:55 01/04/2011.

[statistics] connecting to socket on port 3580
[statistics] connected
.--+----------------+----------+---------------------+-----------+------.
|                                tLogRow_1                                |
|=-+----------------+----------+---------------------+-----------+------=|
|ID|Name            |BirthDate |Address              |City       |Revenue|
|=-+----------------+----------+---------------------+-----------+------=|
|1 |Calvin Wilson   |06-04-2008|997 East Calle Primera|Topeka     |9385   |
|2 |Benjamin Johnson|12-01-2007|386 Newbury Road     |Olympia    |8418   |
|3 |Millard Hayes   |03-05-2007|413 San Marcos       |Carson City|1663   |
|4 |George Quincy   |01-05-2008|516 Carpinteria South|Bismarck   |1997   |
|5 |Ronald Coolidge |22-01-2008|780 Newbury Road     |Jackson    |5515   |
`--+----------------+----------+---------------------+-----------+------'
.--+-----------------+----------+----------------------+-----------+------.
|                                 tLogRow_2                                |
|=-+-----------------+----------+----------------------+-----------+------=|
|ID|Name             |BirthDate |Address               |City       |Revenue|
|=-+-----------------+----------+----------------------+-----------+------=|
|1 |Calvin Wilson    |06-04-2008|997 East Calle Primera|Topeka     |9385   |
|2 |Benjamin Johnson |12-01-2007|386 Newbury Road      |Olympia    |8418   |
|3 |Millard Hayes    |03-05-2007|413 San Marcos        |Carson City|1663   |
|4 |Abraham Roosevelt|09-06-2007|792 South Highway     |Trenton    |3200   |
|5 |Harry Madison    |22-05-2008|246 North Preisker Lane|Honolulu  |2963   |
|6 |Rutherford Reagan|10-12-2008|356 San Ysidro Blvd   |Carson City|2432   |
|7 |Dwight Hayes     |21-03-2007|114 Jean de la Fontaine|Providence|7431   |
|8 |Dwight Taft      |18-08-2007|326 Via Real          |Helena     |8807   |
`--+-----------------+----------+----------------------+-----------+------'

[statistics] disconnected
Job MergeData ended at 10:55 01/04/2011. [exit code=0]
```

tSQLTemplateRollback

Propriétés du tSQLTemplateRollback

Famille du composant	ELT/SQLTemplate	
Fonction	Le **tSQLTemplateRollback** annule la transaction dans une base de données connectée.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Database Type*	Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.
	Component List	Sélectionnez le composant ELT de connexion le plus pertinent dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
SQL Template	*SQLTemplate List*	Cliquez sur le bouton **Add** pour ajouter un ou plusieurs modèle SQL du système à la liste **SQLTemplateList**. Cliquez dans la cellule de la colonne **SQLTemplate List** que vous venez d'ajouter et sélectionnez le modèle SQL du système adéquat dans la liste de modèles fournis par **Talend**. **Note :** Vous pouvez créer vos propres modèles SQL et les ajouter à la liste **SQLTemplate List**. Pour créer un nouveau modèle SQL : 1- Sélectionnez un modèle déjà existant dans la liste **SQLTemplate list** et cliquez dans son code dans le champ de code. Une boîte de dialogue s'ouvrira vous demandant de créer un nouveau modèle. 2- Cliquez sur **Yes** pour ouvrir la boîte de dialogue **[New SQLTemplate]**. 3- Renseignez les informations de votre modèle SQL et cliquez sur **Finish** pour fermer la boîte de dialogue. Un éditeur de modèle SQL apparaît dans lequel vous pouvez saisir votre code. 4- Cliquez sur le bouton **Add** pour ajouter le nouveau modèle dans la liste **SQLTemplate list**. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête SQL traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en association avec des composants ELT, notamment avec le **tSQLTemplateCommit** et le composant ELT de connexion correspondant à la base de données avec laquelle vous travaillez.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé au **tSQLTemplateRollback**, consultez *Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD*.

tSybaseConnection

Le composant **tSybaseConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tSybaseConnection*.

tTeradataConnection

Le composant **tTeradataConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tTeradataConnection*.

tVectorWiseConnection

Le composant **tVectorWiseConnection** appartient à deux familles différentes : Databases et ELT. Pour plus d'informations, consultez *tVectorWiseConnection*.

Composants ESB

Ce chapitre passe en revue les principaux composants de la famille **ESB** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille de composants ESB regroupe les composants dédiés aux tâches relatives à l'ESB.

tESBConsumer

Propriétés du tESBConsumer

Famille de composant	ESB/Web Services	
Fonction	Appelle la méthode définie du Service Web invoqué et retourne la classe, selon les paramètres spécifiés.	
Objectif	Invoque une méthode via un Service Web.	
Basic settings	*Service Configuration*	Description des bindings et de la configuration du Service Web. Le champ **Endpoint** est automatiquement rempli après complétion de la configuration du service.
	Input schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** afin de récupérer automatiquement les colonnes du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Response schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Fault Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Use Service Locator	**Use Service Locator** : Maintient la disponibilité du service pour répondre aux demandes et aux accords de niveau de service (Service Level Agreements). Cette option est indisponible lorsque la case **Use Service Registry** est cochée.
	Use Service Activity Monitor	Capture les événements et stocke les informations pour permettre des analyses en profondeur de l'activité du service et un suivi des messages à travers une transaction métier. Cette option peut être utilisée, entre autres, pour analyser le temps de réponse du service, identifier les modèles de trafic ou effectuer une analyse de cause racine. Cette option est désactivée lorsque la case Use Service Registry est cochée, si vous avez souscrit à l'une des solutions Enterprise ESB de **Talend**.
	Use Authentication	Cochez cette case pour activer l'option d'authentification. Choisissez entre **Basic HTTP**, **HTTP Digest**, **Username Token** et **SAML Token (ESB runtime only)**. Saisissez un identifiant et un mot de passe dans les champs correspondants. Les options **Basic HTTP**, **HTTP Digest** et **Username Token** fonctionnent dans le studio et dans le conteneur d'exécution. L'authentification via **SAML Token** fonctionne uniquement dans le conteneur d'exécution. Lorsque l'option **SAML Token (ESB runtime only)** est sélectionnée, vous pouvez fournir les informations d'authentification pour envoyer la requête ou effectuer l'appel

		si un utilisateur est authentifié, en propageant les informations d'authentification. Sélectionnez : **-** : Saisissez l'identifiant et le mot de passe dans les champs correspondants afin d'accéder au service. **Propagate using U/P** : Saisissez l'identifiant et le mot de passe utilisés pour vous authentifier via STS. **Propagate using Certificate** : Saisissez l'alias et le mot de passe utilisés pour vous authentifier via STS. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Cette option est indisponible lorsque la case **Use Service Registry** est cochée.
	Use Business Correlation	Cochez cette case pour créer un ID de corrélation dans ce composant. Vous pouvez spécifier un ID de corrélation dans le champ **Correlation Value**. Dans ce cas, l'ID de corrélation est passé au service qu'il appelle, afin que les appels en chaîne de services soient groupés sous cet ID de corrélation. Si vous laissez ce champ vide, cette valeur est générée automatiquement au moment de l'exécution. Lorsque cette option est activée, le **tESBConsumer** extrait également l'ID de corrélation de l'en-tête de réponse et le stocke dans le composant pour usage ultérieur dans le flux. Cette option est automatiquement activée lorsque la case **Use Service Registry** est cochée.
	Die on error	Cochez cette case si vous souhaitez arrêter le Job en cas d'erreur.
Advanced settings	*Log messages*	Cochez cette case pour enregistrer l'échange de messages entre le service et le consommateur.
	Service Locator Custom Properties	Cette table apparaît lorsque la case **Use Service Locator** est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ **Property Name** et **Property Value** respectivement.
	Service Activity Custom Properties	Cette table apparaît lorsque la case **Use Service Activity Monitor** est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ **Property Name** et **Property Value** respectivement.
	Connection time out (second)	Configurez une valeur en secondes avant interruption de la connexion au Service Web. Cette option fonctionne uniquement dans le studio. Pour l'utiliser après le déploiement du composant dans le moteur d'exécution : 1. Créez un fichier de configuration avec le nom `org.apache.cxf.http.conduits-<endpoint_name>.cfg` dans le dossier `<TalendRuntimePath>/container/etc/`. 2. Spécifiez l'`url` du service Web et le paramètre `client.ConnectionTimeout` de délai avant suspension en millisecondes dans le fichier de configuration. Si vous devez utiliser l'option **Receive time out**, spécifiez le délai `client.ReceiveTimeout` en millisecondes également. L'`url` peut être un endpoint ou une expression régulière contenant des caractères de remplacement, par exemple :

		`url = http://localhost:8040/*`
		`client.ConnectionTimeout=10000000`
		`client.ReceiveTimeout=20000000`
		
		, dans laquelle `http://localhost:8040/*` correspond à toutes les URL commençant par `http://localhost:8040/`.

	Receive time out (second)	Configurez une valeur en secondes pour la réception de la réponse. Cette option fonctionne uniquement dans le studio. Pour l'utiliser après le déploiement du composant dans le moteur d'exécution, consultez l'option **Connection time out**.
	Trust Server with SSL/ TrustStore file et *TrustStore password*	Cochez cette case pour authentifier le serveur auprès du client via un protocole sécurisé SSL et renseignez les champs suivants : **TrustStore file** : saisissez le chemin d'accès et le nom du fichier TrustStore contenant la liste des certificats approuvés par le client. **TrustStore password** : saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore.
	Use http Proxy/Proxy host, *Proxy port*, *Proxy user* et *Proxy password*	Cochez cette case si vous utilisez un serveur proxy et renseignez les informations nécessaires. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table et renseignez le champ **Code** avec une variable de contexte, afin d'activer ou désactiver dynamiquement l'option **Use Authentication** ou **Use HTTP proxy** lors de l'exécution. Vous pouvez ajouter deux lignes à la table afin de configurer les deux options. Une fois qu'un paramètre dynamique est défini, l'option correspondante est mise en relief et devient inutilisable dans les vues **Basic settings** et **Advanced settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **CORRELATION_ID** : l'ID de corrélation par lequel regrouper les appels de services en chaîne. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé comme composant intermédiaire. Il nécessite un composant d'entrée et un composant de sortie.	
Limitation	Une JDK est requise pour que ce composant puisse fonctionner.	

Scénario 1 : Utiliser le composant tESBConsumer

Ce scénario décrit un Job utilisant un composant **tESBConsumer** pour définir si des adresses e-mail sont valides ou non.

Construire le Job

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tFixedFlowInput**, un **tXMLMap**, un **tESBConsumer** et deux **tLogRow**.
2. Cliquez-droit sur le composant **tFixedFlowInput**, sélectionnez **Row** > **Main** dans le menu contextuel et cliquez sur le **tXMLMap**.
3. Cliquez-droit sur le composant **tXMLMap**, sélectionnez **Row** > ***New Output* (Main)** dans le menu contextuel et cliquez sur le **tESBConsumer**. Saisissez *payload* dans la boîte de dialogue afin de nommer ce lien et acceptez la propagation afin de récupérer le schéma du composant **tESBConsumer**.
4. Cliquez-droit sur le **tESBConsumer**, sélectionnez **Row** > **Response** dans le menu contextuel et cliquez sur l'un des **tLogRow**.
5. Cliquez-droit sur le composant **tESBConsumer**, sélectionnez **Row** > **Fault** dans le menu contextuel et cliquez sur le second **tLogRow**.

Configurer les composants

Configurer le tFixedFlowInput

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit Schema**. Dans la boîte de dialogue du schéma, cliquez sur le bouton **[+]** pour ajouter une nouvelle ligne de type **String** et nommez-la *payloadString*. Cliquez sur **OK** pour fermer la boîte de dialogue.

3. Dans le champ **Number of rows**, configurez le nombre de lignes à *1*.

4. Dans la zone **Mode**, sélectionnez **Use Single Table** et saisissez la requête suivante entre guillemets doubles dans le champ **Value** :

```
nomatter@gmail.com
```

Configurer le tXMLMap

1. Dans l'espace de modélisation graphique, double-cliquez sur le composant **tXMLMap** pour ouvrir son éditeur **Map Editor**.

2. Dans la partie inférieure droite de l'éditeur, cliquez sur le bouton **[+]** pour ajouter une ligne de type `Document` à la table de sortie et nommez-la *payload*.

3. Dans la table de sortie, cliquez-droit sur le nœud racine et sélectionnez **Rename** dans le menu contextuel. Saisissez *IsValidEmail* dans la boîte de dialogue qui apparaît.

4. Cliquez-droit sur le nœud *IsValidEmail* et sélectionnez **Set A Namespace** dans le menu contextuel. Saisissez *http://www.webservicex.net* dans la boîte de dialogue qui apparaît.

5. Cliquez-droit sur le nœud *IsValidEmail* à nouveau et sélectionnez **Create Sub-Element**. Saisissez *Email* dans la boîte de dialogue.

6. Cliquez-droit sur le nœud *Email* et sélectionnez **As loop element** dans le menu contextuel.

7. Cliquez sur le nœud *payloadString* de la table d'entrée et déposez-le dans la colonne **Expression**, dans la ligne du nœud *Email* dans la table de sortie.

8. Cliquez sur **OK** pour valider le mapping et fermer l'éditeur **Map Editor**.

Configurer le tESBConsumer

1. Dans l'espace de modélisation graphique, double-cliquez sur le **tESBConsumer** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Service configuration**.

3. Dans la boîte de dialogue qui apparaît, saisissez : *http://www.webservicex.net/ValidateEmail.asmx?WSDL* dans le champ **WSDL** et cliquez sur le bouton **Refresh** pour récupérer le nom du port et de l'opération. Dans la liste **Port Name**, sélectionnez le port à utiliser, *ValidateEmailSoap* dans cet exemple. Cliquez sur **Finish** pour valider vos paramètres et fermer la boîte de dialogue.

Les **tLogRow** monitorent les échanges de message et ne nécessitent pas de configuration. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

Exécuter le Job

Cliquez sur l'onglet **Run** pour l'afficher et cliquez sur le bouton **Run** afin de lancer l'exécution du Job. Vous pouvez également appuyer sur **F6** pour l'exécuter. Dans la console, vous pouvez voir :

Scénario 2 : Utiliser le tEBSConsumer avec des en-têtes SOAP personnalisés

Ce scénario est similaire au précédent. Le Job utilise un **tESBConsumer** pour récupérer une adresse e-mail valide ainsi que des en-têtes personnalisés SOAP dans le message de requête.

Déposer et relier les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tESBConsumer**, un **tMap**, deux **tFixedFlowInput**, deux **tXMLMap** et deux **tLogRow**.

2. Reliez les composants **tFixedFlowInput** au **tXMLMap** à l'aide d'un lien **Row** > **Main**.

3. Cliquez-droit sur le premier **tXMLMap**, sélectionnez **Row** > ***New Output* (Main)** dans le menu contextuel et cliquez sur le **tMap**. Saisissez *payload*dans la boîte de dialogue qui s'ouvre, pour nommer cette connexion.

   Répétez l'opération pour relier le second **tXMLMap** au **tMap** et nommez le lien de sortie *header*.

4. Cliquez-droit sur le composant **tMap**, sélectionnez **Row** > ***New Output* (Main)** dans le menu contextuel et cliquez sur le composant **tESBConsumer**. Saisissez *request* dans la boîte de dialogue qui s'ouvre, afin de nommer cette ligne et acceptez la propagation proposée afin d'obtenir le schéma du **tESBConsumer**.

5. Cliquez-droit sur le **tESBConsumer**, sélectionnez **Row** > **Response** dans le menu contextuel et cliquez sur l'un des **tLogRow**.

6. Cliquez-droit sur le **tESBConsumer** à nouveau, sélectionnez **Row** > **Fault** dans le menu contextuel puis cliquez sur l'autre **tLogRow**.

Configurer les composants

Configurer le composant tFixedFlowInput

1. Double-cliquez sur le premier **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit Schema**. Dans la boîte de dialogue du schéma, cliquez sur le bouton **[+]** pour ajouter une ligne de type **String** et nommez-la *payload*. Cliquez sur **OK** pour fermer la boîte de dialogue.

3. Dans le champ **Number of rows**, paramétrez le nombre de lignes à *1*.

4. Dans la zone **Mode**, sélectionnez **Use Single Table** et saisissez `"nomatter@gmail.com"` dans le champ **Value**, le payload du message de requête.

5. Configurer le second **tFixedFlowInput** de la même façon, mais créez un schéma différent.

   Dans le schéma, ajoutez deux lignes de type **String** et nommez-les *id* et *company*, respectivement.

Donnez la valeur *Hello world!* à la colonne *id* et *Talend* à la colonne *company*, qui sont les en-têtes du message de requête.

Configurer les composants tXMLMap

1. Dans l'espace de modélisation graphique, double-cliquez sur le premier **tXMLMap** pour ouvrir son éditeur **Map Editor**.

2. Dans la partie inférieure droite de l'éditeur, cliquez sur le bouton **[+]** pour ajouter une ligne de type `Document` dans la table de sortie. Nommez-la *payload*.

3. Dans la table de sortie, cliquez-droit sur le nœud racine et sélectionnez **Rename** dans le menu contextuel. Saisissez *IsValidEmail* dans la boîte de dialogue qui s'ouvre.

4. Cliquez-droit sur le nœud *IsValidEmail* et sélectionnez **Set A Namespace** dans le menu contextuel. Saisissez *http://www.webservicex.net* dans la boîte de dialogue qui s'ouvre.

5. Cliquez-droit à nouveau sur le nœud *IsValidEmail* et sélectionnez **Create Sub-Element** dan le menu contextuel. Saisissez *Email* dans la boîte de dialogue qui s'ouvre.

6. Cliquez-droit sur le nœud *Email* et sélectionnez **As loop element** dans le menu contextuel.

7. Cliquez sur le nœud *payload* dans la table d'entrée et déposez-le dans la colonne **Expression** de la ligne de nœud *Email*, dans la table de sortie.

8. Cliquez sur **OK** pour valider le mapping et fermer l'éditeur **Map Editor**.

9. Configurez de la même façon l'autre **tXMLMap**. Ajoutez une ligne de type `Document` à la table de sortie et nommez-la *header*. Créez-lui deux sous-éléments, *id* et *company*. Mappez les nœuds *id* et *company* de la table d'entrée avec les nœuds correspondants dans la table de sortie.

Configurer le composant tMap

1. Dans l'espace de modélisation graphique, double-cliquez sur le **tMap** pour ouvrir son **Map Editor**.

2. Dans la partie inférieure droite de l'éditeur, cliquez sur le bouton **[+]** pour ajouter deux lignes de type `Document` à la table de sortie et nommez-les respectivement *payload* et *headers*.

3. Cliquez sur le nœud *payload* dans la table d'entrée et déposez-le dans la colonne **Expression** de la ligne du nœud *payload*, dans la table de sortie.

4. Cliquez sur le nœud *header* dans la table d'entrée et déposez-le dans la colonne **Expression** de la ligne du nœud *headers* dans la table de sortie.

Configurer le composant tESBConsumer

1. Dans l'espace de modélisation graphique, double-cliquez sur le **tESBConsumer** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Service configuration**.

3. Dans la boîte de dialogue qui s'ouvre, saisissez : *http://www.webservicex.net/ValidateEmail.asmx?WSDL* dans le champ **WSDL** et cliquez sur le bouton **Refresh** (flèches) afin de récupérer le nom du port et le nom de l'opération. Dans la liste **Port Name**, sélectionnez le port à utiliser, *ValidateEmailSoap* dans cet exemple. Cliquez sur **OK** pour valider les paramètres et fermer la boîte de dialogue.

4. Dans la vue **Advanced settings**, cochez la case **Log messages** pour enregistrer le contenu des messages.

Les composants **tLogRow** monitorent les échanges de messages et ne nécessitent aucune configuration particulière. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

Exécuter le Job

Cliquez sur l'onglet **Run** pour afficher la vue du même nom. Cliquez sur le bouton **Run** pour lancer l'exécution de votre Job. Vous pouvez également appuyer sur **F6** pour l'exécuter.

```
Execution
Run   Kill   Clear

[statistics] connecting to socket on port 3768
[statistics] connected
15:24:06,352 INFO  [ReflectionServiceFactoryBean] Creating Service
{http://www.webservicex.net}ValidateEmail from WSDL:
classpath:/esb_demo/tesbconsumer_soapheader_0_1/wsdl/tESBConsumer_1.wsdl
15:24:06,508 INFO  [ValidateEmailSoap] Outbound Message
---------------------------
ID: 1
Address: http://www.webservicex.net/ValidateEmail.asmx
Encoding: UTF-8
Http-Method: POST
Content-Type: application/soap+xml; action="http://www.webservicex.net/IsValidEmail"
Headers: {Accept=[*/*]}
Payload: <soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"><soap:Header><id>Hello
world!</id><company>Talend</company></soap:Header><soap:Body><IsValidEmail
xmlns="http://www.webservicex.net"><Email>nomatter@gmail.com</Email></IsValidEmail></soap:Body>
</soap:Envelope>
--------------------------------------
15:24:28,443 INFO  [ValidateEmailSoap] Inbound Message
---------------------------
ID: 1
Response-Code: 200
Encoding: UTF-8
Content-Type: application/soap+xml; charset=utf-8
Headers: {Cache-Control=[private, max-age=0], Content-Length=[370], content-type=
[application/soap+xml; charset=utf-8], Date=[Tue, 24 Sep 2013 07:24:44 GMT], Server=
[Microsoft-IIS/7.0], X-AspNet-Version=[4.0.30319], X-Powered-By=[ASP.NET]}
Payload: <?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><IsValidEmailResponse
xmlns="http://www.webservicex.net"><IsValidEmailResult>false</IsValidEmailResult></IsValidEmail
Response></soap:Body></soap:Envelope>
--------------------------------------
<?xml version="1.0" encoding="UTF-8"?>
<IsValidEmailResponse
xmlns="http://www.webservicex.net"><IsValidEmailResult>false</IsValidEmailResult></IsValidEmail
Response>
[statistics] disconnected

Line limit  100   Wrap
```

Comme affiché dans le log d'exécution, l'en-tête SOAP est passé au service, dans la requête.

tESBProviderFault

*Ce composant est adapté pour une utilisation au sein de la perspective **Mediation** du Studio Talend. Il requiert l'utilisation du nœud du Repository **Service** et des assistants de création de **Services**.*

Propriétés du tESBProviderFault

Famille de composant	ESB/Web Services	
Fonction	Génère un message Fault (message d'erreur) provenant du Service Web, à la fin du Job **Talend**.	
Objectif	Retourne un message Fault provenant du Service Web à la fin du cycle du Job **Talend**.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** afin de récupérer automatiquement les colonnes du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	EBS service settings	**Fault title** : Valeur de la colonne *faultString* dans le message Fault. Le flux **Row** > **Fault** du composant **tESBConsumer** contient un schéma prédéfini dont la colonne faultString contient les données du champ **Fault title** du **tESBProviderFault**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant doit être utilisé uniquement avec le composant **tESBProviderRequest**.
Limitation	Une JDK est requise pour que ce composant puisse fonctionner.

Scénario : Effectuer une requête sur les nom d'aéroports à partir des codes pays

Ce scénario contient deux Jobs, un fournisseur de service de données et un consommateur de service de données. Le premier Job écoute les requêtes du consommateur via le **tESBProviderRequest**, met en correspondance le code pays contenu dans la requête avec une table d'une base de données MySQL contenant des paires code pays/aéroport via le **tMap** puis retourne le nom correct de l'aéroport via le **tESBProviderResponse**. Si aucune correspondance n'est trouvée, un message d'erreur est retourné via le **tESBProviderFault**. Le consommateur envoie des requêtes au fournisseur et reçoit les informations concernant les aéroports ou les erreurs via le **tESBConsumer**.

Construire le fournisseur de service de données pour publier un service

Le service de données *airport* a déjà été configuré sous le nœud **Services** de la vue **Repository**. Son schéma contient trois éléments majeurs :

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soapenc="http://s
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://airportsoap.sopera.de">
      <s:element name="getAirportInformationByISOCountryCode">
      <s:element name="getAirportInformationByISOCountryCodeResponse">
      <s:element name="getAirportInformationByISOCountryCodeFault">
        <s:complexType>
```

Pour plus d'informations concernant la définition d'un service dans le studio, consultez le *Guide utilisateur du Studio Talend*.

Assigner un Job au service défini

1. Cliquez-droit sur *getAirportInformationByISOCountryCode* sous le service Web *airport* et, dans le menu contextuel, sélectionnez **Assign Job**.

2. Dans la fenêtre **[Operation Choice]**, sélectionnez **Create a new Job and Assign it to this Service Operation**.

3. Cliquez sur **Next** pour ouvrir la fenêtre de description de Job. Le nom du Job *airportSoap_getAirportInformationByISOCountryCode* est automatiquement renseigné.

4. Cliquez sur **Finish** pour créer le Job et l'ouvrir dans l'espace de modélisation graphique. Trois composants sont déjà disponibles.

Déposer et relier les composants

1. Déposez un **tXMLMap** et un **tMysqlInput** de la **Palette** dans l'espace de modélisation graphique.
2. Reliez le **tESBProviderRequest** au **tXMLMap** à l'aide d'un lien **Row** > **Main**.
3. Reliez le **tMysqlInput** au **tXMLMap** à l'aide d'un lien **Row** > **Main**.
4. Reliez le **tXMLMap** au **tESBProviderResponse** à l'aide d'un lien **Row** > ***New Output*(Main)**.

   Dans la fenêtre **[new Output name]**, saisissez le nom de la table de sortie, *airport_response*.

   Cliquez sur **OK** dans la fenêtre vous proposant de récupérer le schéma du composant cible.
5. Reliez le **tXMLMap** au **tESBProviderFault** à l'aide d'un lien **Row** > ***New Output*(Main)**.

   Dans la fenêtre **[new Output name]** qui s'ouvre, saisissez le nom de la table de sortie, *fault_message*.

   Cliquez sur **OK** dans la fenêtre vous proposant de récupérer le schéma du composant suivant.

Configurer les composants

1. Double-cliquez sur le **tMysqlInput** pour afficher sa vue **Basic settings**.

2. Renseignez les propriétés de base de la connexion MySQL et de la table de base de données.

   Cliquez sur le bouton **[...]** pour ouvrir l'éditeur de schéma.

3. Cliquez deux fois sur le bouton **[+]** pour ajouter deux colonnes de type String. Nommez-les respectivement *id* et *name*.

   Cliquez sur **OK** pour fermer l'éditeur.

   Cliquez sur **Guess Query** pour récupérer la requête SQL.

4. Double-cliquez sur le **tXMLMap** pour ouvrir l'éditeur de mapping.

5. Dans la table **main : row1** du côté de l'entrée (gauche), cliquez-droit sur le nom de la colonne *payload* et, dans le menu contextuel, sélectionnez **Import from Repository**. L'assistant **[Metadata]** s'ouvre.

Sélectionnez le schéma du message de requête et cliquez sur **OK** pour valider la sélection. Dans cet exemple, le schéma est *getAirportInformationByISOCountryCode*.

6. Répétez l'opération pour importer le schéma hiérarchique pour les messages Response/Fault (droite). Dans cet exemple, les schémas sont respectivement *getAirportInformationByISOCountryCodeResponse* et *getAirportInformationByISOCountryCodeFault*.

7. Pour créer la jointure sur les données de référence, déposez le nœud *CountryAbbrviation* de la table principale sur la ligne *id* de la table de référence (lookup).

8. Dans la table de référence, cliquez sur l'icône de clé anglaise, dans le coin supérieur droit, pour ouvrir le panneau des paramètres.

   Paramétrez **Lookup Model** à **Reload at each row**, **Match Model** à **All matches** et **Join Model** à **Inner join**.

9. Dans la table du flux de sortie *airport_response*, cliquez sur l'icône de clé anglaise, dans le coin supérieur droit, pour ouvrir le panneau des paramètres.

   Paramétrez l'option **All in one** à **true**. Cela vous assure qu'une seule réponse est retournée pour chaque requête si plusieurs aéroports correspondent dans la base de données.

10. Dans la table de sortie *fault_message*, cliquez sur l'icône de clé anglaise, dans le coin supérieur droit, pour ouvrir le panneau des paramètres.

    Paramétrez l'option **Catch Lookup Inner Join Reject** à **true** pour monitorer les non-correspondances entre le code pays de la requête et les enregistrements dans la table de base de données. Lorsqu'une telle situation se produit, un message Fault est généré par le **tESBConsumer** et écrit en sortie par le flux **Row** > **Fault**.

    Le flux **Row** > **Fault** du **tESBConsumer** contient un schéma prédéfini dont la colonne *faultString* contient les données du champ **Fault title** du **tESBProviderFault**.

11. Déposez la colonne *name* du flux de référence dans la zone **Expression** dans la table de sortie *airport_response*, à côté du nœud *tns:getAirportInformationByISOCountryCodeResult*.

    Déposez le nœud *tns:CountryAbbreviation* du flux principal dans la zone **Expression** à côté du nœud *tns:getAirportInformationByISOCountryCodeFaultString* dans la table de sortie *fault_message*. Ainsi, le code pays incorrect de la requête s'affiche dans la colonne *faultDetail* du flux **Row** > **Fault** du **tESBConsumer**.

    Cliquez sur **OK** pour fermer l'éditeur et valider la configuration.

12. Double-cliquez sur le **tESBProviderFault** pour afficher sa vue **Basic settings**.

13. Dans le champ **Fault title**, saisissez la variable de contexte *context.fault_message*.

    Pour plus d'informations concernant la définition des variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

Exécuter le Job

1. Appuyez sur les touches **Ctrl +S** pour sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter le Job.

```
[statistics] connecting to socket on port 3876
[statistics] connected
[INFO ]: org.apache.cxf.service.factory.ReflectionServiceFactoryBean -
Creating Service {http://airportsoap.sopera.de}airport from WSDL:
E:/Talend/Talend-Studio-r110103-V5.4.1NB/workspace/PROJECT_20131029/services/
airport_0.1.wsdl
[INFO ]: org.apache.cxf.endpoint.ServerImpl - Setting the server's publish
address to be http://localhost:8200/airport.service
[INFO ]: org.eclipse.jetty.server.Server - jetty-8.1.12.v20130726
[INFO ]: org.eclipse.jetty.server.AbstractConnector - Started
SelectChannelConnector@localhost:8200
web service [endpoint: http://localhost:8200/airport.service] published
```

Le service de données est publié et écoute toutes les requêtes jusqu'à ce que vous cliquiez sur le bouton **Kill** pour l'arrêter, puisque, par défaut, l'option **Keep listening** du **tESBProviderRequest** est automatiquement sélectionnée.

Configurez le Job consommateur qui interagit avec le service de données.

Construire le consommateur de service de données pour envoyer une requête au service

Construit autour du **tESBConsumer**, le Job consommateur envoie deux requêtes contenant des codes pays au service Web pour obtenir le nom des aéroports correspondants. Si un mauvais code pays est contenu dans la requête, un message d'erreur est retourné. Les codes pays et les enregistrements de la base de données MySQL se présentent comme suit :

id	name
CN	Capital Airport
DE	Frankfurt Airport
FR	Roissy Airport
* (NULL)	(NULL)

country_code.txt

1	CC
2	CN

Déposer et relier les composants

1. Déposez un **tFileInputDelimited**, un **tXMLMap**, un **tESBConsumer** et deux **tLogRow** de la **Palette** dans l'espace de modélisation graphique.

2. Renommez un **tLogRow** *response* et l'autre *fault_message*.

3. Reliez le **tFileInputDelimited** au **tXMLMap** à l'aide d'un lien **Row** > **Main**.

4. Reliez le **tXMLMap** au **tESBConsumer** à l'aide d'un lien **Row** > ***New Output*(Main)**.

   Dans la fenêtre **[new Output name]** qui s'ouvre, saisissez le nom de la table de sortie, par exemple *request*.

   Cliquez sur **OK** dans la fenêtre vous proposant de récupérer le schéma du composant cible.

5. Reliez le **tESBConsumer** au composant **response** à l'aide d'un lien **Row** > **Response**.

6. Reliez le **tESBConsumer** au **fault_message** à l'aide d'un lien **Row** > **Fault**.

Configurer les composants

1. Double-cliquez sur le **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **File name/stream**, saisissez la variable de contexte pour le fichier contenant les codes pays, par exemple *context.filepath*.

3. Cliquez sur le bouton **[...]** pour ouvrir l'éditeur de schéma.

4. Cliquez sur le bouton **[+]** pour ajouter une colonne *country_code*, de type String.

   Cliquez sur **OK** pour fermer l'éditeur.

5. Double-cliquez sur le **tXMLMap** pour ouvrir son éditeur de mapping.

6. Dans la table *request* du flux de sortie, cliquez-droit sur la colonne *payload* et, dans le menu contextuel, sélectionnez **Import from Repository**. L'assistant **[Metadata]** s'ouvre.

Sélectionnez le schéma du message de requête et cliquez sur **OK** pour valider la sélection. Dans cet exemple, le schéma est *getAirportInformationByISOCountryCode*.

7. Déposez la colonne *country_code* du flux principal dans la zone **Expression**, à côté du nœud *tns:CountryAbbreviation* dans le flux de sortie *request*.

Cliquez sur **OK** pour fermer l'éditeur et valider la configuration.

8. Double-cliquez sur le composant **tESBConsumer** pour ouvrir son assistant de configuration de service.

9. Cliquez sur le bouton **Browse...** pour sélectionner le fichier WSDL défini. Les champs **Port name** et **Operation** sont automatiquement renseignés, une fois le fichier WSDL sélectionné.

   Cliquez sur **OK** pour fermer l'assistant.

10. Double-cliquez sur le composant **response** pour ouvrir sa vue **Basic settings**.

11. Sélectionnez l'option **Vertical (each row is a key/value list)** puis **Print label** afin de visualiser les résultats de manière optimale.

    Répétez l'opération pour l'autre **tLogRow**, **fault_message**.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter ce Job.

```
#1. fault_message
|
+-------------
+-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------+
| key          | value
|
+-------------
+-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------+
| faultCode   | {http://airportsoap.sopera.de}businessFault
|
| faultString | Country Code Incorrect [tESBProviderFault_1]
|
| faultActor  | null
|
| faultNode   | null
|
| faultRole   | null
|
| faultDetail | <?xml version="1.0" encoding="UTF-8"?>
<tns:getAirportInformationByISOCountryCodeFault
xmlns:tns="http://airportsoap.sopera.de"><tns:getAirportInformation
ByISOCountryCodeFaultString>CC</tns:getAirportInformationByISOCount
ryCodeFaultString></tns:getAirportInformationByISOCountryCodeFault>
|
```

```
#1. response
|
+---------
+-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------------------------------------------------
-------------------------------+
| key      | value
|
+---------
+-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------------------------------------------------
-------------------------------+
| payload | <?xml version="1.0" encoding="UTF-8"?>
<tns:getAirportInformationByISOCountryCodeResponse
xmlns:tns="http://airportsoap.sopera.de"><tns:getAirportInformation
ByISOCountryCodeResult>Capital
Airport</tns:getAirportInformationByISOCountryCodeResult></tns:getA
irportInformationByISOCountryCodeResponse> |
+---------
```

Comme affiché ci-dessus, deux messages sont retournés, le premier donnant le nom de l'aéroport correspondant au code pays *CN* et le second donnant les détails de l'erreur causée par le code pays *CC*.

tESBProviderRequest

*Ce composant est adapté pour une utilisation au sein de la perspective **Mediation** du Studio Talend. Il requiert l'utilisation du nœud du Repository **Service** et des assistants de création de **Services**.*

Propriétés du tESBProviderRequest

Famille de composant	ESB/Web Services	
Fonction	Expose un Job **Talend** comme un Service Web.	
Objectif	Attend un message de requête d'un consommateur et le passe au composant suivant.	
Basic settings	*Property Type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Aucun fichier WSDL n'est configuré pour le Job.
		Repository : Sélectionnez le fichier de service Web souhaité depuis le **Repository** pour sélectionner les paramètres (granularité) du nom du port et de l'opération.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Keep listening	Cochez cette case lorsque vous souhaitez assurer que le fournisseur (le Job **Talend**) continue à écouter les requêtes après traitement de la première requête entrante.
Advanced settings	*Log messages (Studio only)*	Cochez cette case pour enregistrer l'échange de messages entre le service et le consommateur. Cette option fonctionne uniquement avec le Studio.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin d'activer ou désactiver dynamiquement l'option **Keep listening** au moment de l'exécution. Lorsqu'un paramètre dynamique est défini, l'option **Keep listening** correspondante est désactivée dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	

Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **CORRELATION_ID** : l'ID de corrélation par lequel regrouper les appels de services en chaîne. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **SECURITY_TOKEN** : informations concernant l'identité de l'utilisateur dans l'en-tête de requête. Cette variable est une variable **Flow** et retourne un nœud XML. **HEADERS_SOAP** : en-têtes de la requête SOAP. Cette variable est une variable **Flow** et retourne tous les en-têtes des requêtes SOAP. **HEADERS_HTTP** : en-têtes de la requête HTTP. Cette variable est une variable **Flow** et retourne tous les en-têtes des requêtes HTTP. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant couvre la possibilité d'exposer un Job **Talend** en tant que Service, avec la possibilité d'écrire une requête de service dans un Job et retourne le résultat du Job en tant que réponse. Le composant **tESBProviderResponse** peut livrer le payload d'un message SOAP et accéder aux en-têtes HTTP et SOAP d'un service. Le composant **tESBProviderRequest** doit être utilisé avec le **tESBProviderResponse** afin de fournir une réponse à la fin du Job, dans le cas d'une communication de style requête-réponse. Lorsque SAML Token ou Service Registry est activé dans les options du Runtime et si le jeton SAML existe dans l'en-tête de la requête, le **tESBProviderRequest** obtient et stocke le jeton SAML dans la variable du composant pour un usage ultérieur dans le flux. Le **tESBProviderRequest** obtient une valeur de corrélation (Correlation Value) dans l'en-tête de la requête si la valeur existe et qu'elle est stockée dans la variable du composant. Lorsqu'une corrélation métier (Business Correlation) ou le registre de Service (Service Registry) est activé(e) dans les options du Runtime, la Correlation Value est également ajoutée à la réponse. Dans ce cas, le **tESBProviderRequest** crée une Correlation Value si elle n'existe pas. Notez que l'option de Service Registry est disponible uniquement si vous avez souscrit à l'une des solutions Enterprise ESB de **Talend**. Pour plus d'informations concernant les options du Runtime, consultez la section correspondante dans le *Guide utilisateur du Studio Talend*.
Limitation	Une JDK est requise pour que ce composant fonctionne.

Scénario : Envoi d'un message via un service sans attente de réponse

Les Jobs suivants, construits sur les composants de la famille **ESB/Web Service** implémentent des services Web définis dans le nœud **Services** du **Repository**. Ils requièrent la création et l'association des services correspondants. Pour plus d'informations concernant les services, consultez le *Guide utilisateur du Studio Talend*.

Dans ce scénario, un Job fournisseur et un Job consommateur sont nécessaires. Par ailleurs, le service correspondant doit déjà exister sous le nœud **Services**, avec l'URI *http://127.0.0.1.8088/esb/provider/?WSDL*, le port *TEST_ProviderJobSoapBinding* et l'opération *invoke(anyType):anyType*

Le Job fournisseur comprend les composants suivants : **tESBProvider**, **tXMLMap** et deux **tLogRow**.

- Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tESBProviderRequest**, un **tXMLMap** et deux **tLogRow**.
- Double-cliquez sur le composant **tESBProvider Request** afin d'ouvrir sa vue **Basic settings**.

- Sélectionnez **Repository** dans la liste **Property Type** et cliquez sur le bouton **[...]** afin de choisir le service, la granularité du port et l'opération.

- Cliquez sur **OK**.
- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour voir le schéma du composant **tESBProvider Request**.

- Cliquez sur **OK**.
- Connectez le **tESBProviderRequest** au composant **tLogRow_1**.
- Double-cliquez sur le **tLogRow_1** dans l'espace de modélisation graphique afin d'afficher sa vue **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et configurez le schéma comme suit.

- Connectez le **tLogRow_1** au **tXMLMap**.
- Reliez le **tXMLMap** et **tLogRow_2** et nommez la connexion *payload*.
- Dans l'espace de modélisation graphique, double-cliquez sur le **tXMLMap** afin d'ouvrir le **Map Editor**.
- Dans la partie inférieure droite de l'éditeur, cliquez sur le bouton **[+]** pour ajouter une ligne à la table *payload* et nommez cette ligne *payload*.
- Dans la colonne **Type** de cette ligne *payload*, sélectionnez le type de données **Document**. La racine XML correspondante est ajoutée automatiquement à la table supérieure de droite, représentant le flux de sortie.
- Dans la table *payload*, cliquez-droit sur *root* pour ouvrir son menu contextuel.
- Dans le menu contextuel, sélectionnez **Create Sub-Element** et saisissez *response* dans la boîte de dialogue.
- Cliquez-droit sur le nœud *response* et sélectionnez **As loop element** dans le menu contextuel.
- Répétez l'opération pour créer un sous-élément *request* du noeud *root* dans la table d'entrée et définissez le nœud *request* comme élément de boucle.
- Cliquez sur le nœud *request* dans la table d'entrée et déposez-le dans la colonne **Expression** de la ligne du nœud *response* de la table de sortie.

- Cliquez sur **OK** pour valider le mapping et fermer l'éditeur.
- Double-cliquez sur le **tLogRow_2** dans l'espace de modélisation graphique afin d'afficher sa vue **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et configurez le schéma comme suit.

- Sauvegardez le Job.

Le Job consommateur comprend un **tFixedFlowInput**, un **tXMLMap**, un **tESBConsumer** et deux **tLogRow**.

- Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tFixedFlowInput**, un **tXMLMap**, un **tESBConsumer** et deux **tLogRow**.

- Double-cliquez sur le **tFixedFlowInput** afin d'afficher sa vue **Basic settings**.

- Configurez le schéma du composant **tFixedFlowInput**.

- Cliquez sur le bouton **[+]** pour ajouter une ligne de type **String** et nommez-la *payloadString*.
- Cliquez sur **OK**.
- Dans le champ **Number of rows**, configurez le nombre de lignes à *1*.
- Dans la zone **Mode**, sélectionnez **Use Single Table** et saisissez `world` entre guillemets, dans le champ **Value**.
- Connectez le **tFixedFlowInput** au **tXMLMap**.
- Reliez le **tXMLMap** au **tESBConsumer** et nommez la connexion *payload*.
- Dans l'espace de modélisation graphique, double-cliquez sur le **tXMLMap** pour ouvrir le **Map Editor**.
- Dans la table de sortie, cliquez-droit sur *root* pour ouvrir son menu contextuel.
- Dans le menu contextuel, sélectionnez **Create Sub-Element** et saisissez *request* dans la boîte de dialogue.
- Cliquez-droit sur le nœud *request* et sélectionnez **As loop element** dans le menu contextuel.
- Cliquez sur le nœud *payloadstring* dans la table d'entrée et déposez-le dans la colonne **Expression** dans la ligne du nœud *request* de la table de sortie.

- Cliquez sur **OK** pour valider le mapping et fermer l'éditeur.

- Démarrez le Job fournisseur. Dans le log d'exécution, vous pouvez voir :

```
...
web service [endpoint: http://127.0.0.1:8088/esb/provider] published
...
```

- Dans la vue **Component** du **tESBConsumer**, configurez ses propriétés.

- Cliquez sur le bouton **[...]** à côté du champ **Service Configuration** afin d'ouvrir l'éditeur.

- Dans le champ **WSDL**, saisissez : *http://127.0.0.1:8088/esb/provider?WSDL*
- Cliquez sur le bouton **Refresh** afin de récupérer le nom du port et de l'opération.
- Cliquez sur **OK**.

- Configurez le schéma d'entrée (**Input Schema**) du composant **tESBConsumer** comme suit.

- Configurez le schéma de réponse (**Response Schema**) comme suit.

- Configurez le schéma **Fault** comme suit.

- Connectez le composant **tESBConsumer** au **tLogRow_1** et au **tLogRow_2**.
- Dans l'espace de modélisation graphique, double-cliquez sur le **tLogRow_1** afin d'afficher sa vue **Basic settings**.

- Cliquez sur le bouton à côté du champ **Edit schema** et configurez le schéma comme suit.

- Dans l'espace de modélisation graphique, double-cliquez sur **tLogRow_2** afin d'afficher sa vue **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et configurez le schéma comme suit.

- Sauvegardez le Job.
- Exécutez le Job fournisseur. Vous pouvez voir dans le log d'exécution :

  INFO: Setting the server's publish address to be http://127.0.0.1:8088/esb/provider

  2011-04-21 14:14:36.793:INFO::jetty-7.2.2.v20101205

  2011-04-21 14:14:37.856:INFO::Started

  SelectChannelConnector@127.0.0.1:8088

  web service [endpoint: http://127.0.0.1:8088/esb/provider] published

- Exécutez le Job consommateur. Vous pouvez voir dans le log d'exécution du Job :

```
Starting job CallProvider at 14:15 21/04/2011.

[statistics] connecting to socket on port 3942
[statistics] connected
TEST_ESBProvider2
TEST_ESBProvider2SoapBingding
|
[tLogRow_2] payloadString: <request>world</request>
{http://talend.org/esb/service/job}TEST_ESBProvider2
{http://talend.org/esb/service/job}TEST_ESBProvider2SoapBinding
invoke
[tLogRow_1] payload: null
[statistics] disconnected
Job CallProvider2 ended at 14:16 21/04/2011. [exit code=0]
```

- Dans le log du fournisseur vous pouvez voir le log des traces :

```
web service [endpoint: http://127.0.0.1:8088/esb/provider]
published
```

```
[tLogRow_1] payload: <?xml version="1.0" encoding="UTF-8"?>
<request>world</request>
### world
[tLogRow_2] content: world
[tLogRow_3] payload: <?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://talend.org/esb/service/job">Hello, world!</response>
web service [endpoint: http://127.0.0.1:8088/esb/provider] unpublished
[statistics] disconnected
Job ESBProvider2 ended at 14:16 21/04/2011. [exit code=0]
```

tESBProviderResponse

*Ce composant est adapté pour une utilisation au sein de la perspective **Mediation** du Studio Talend. Il requiert l'utilisation du nœud du Repository **Service** et des assistants de création de **Services**.*

Propriétés du tESBProviderResponse

Famille de composant	ESB/Web Services	
Fonction	Le composant **tESBProviderResponse** génère un message de réponse à la fin d'un Job **Talend**.	
Objectif	Retourne une réponse comme fournisseur de service à la fin de chaque cycle d'un Job **Talend**.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** afin de récupérer automatiquement les colonnes du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le composant **tESBProviderRequest**. doit être utilisé avec le **tESBProviderRequest** afin de fournir une réponse à la fin d'un Job, pour un fournisseur de service, dans le cas d'une communication de style requête-réponse.	
Limitation	Une JDK est requise pour que ce composant fonctionne.	

Scénario : Retourner une réponse "Hello world"

Les Jobs suivants, construits sur les composants de la famille **ESB/Web Service** implémentent des services Web définis dans le nœud **Services** du **Repository**. Ils requièrent la création et l'association des services correspondants. Pour plus d'informations concernant les services, consultez le *Guide utilisateur du Studio Talend*.

Dans ce scénario, un Job fournisseur et un Job consommateur sont nécessaires. Par ailleurs, le service correspondant doit déjà exister sous le nœud **Services**, avec l'URI *http://127.0.0.1.8088/esb/provider/?WSDL*, le port *TEST_ProviderJobSoapBinding* et l'opération *invoke(anyType):anyType*

Le Job fournisseur comprend les composants suivants : un **tESBProviderRequest**, un **tXMLMap** et deux **tLogRow**.

- Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tESBProviderRequest**, un **XMLMap** et deux **tLogRow**.
- Dans l'espace de modélisation graphique, double-cliquez sur le composant **tESBProviderRequest** afin d'afficher sa vue **Basic settings**.

- Dans la liste **Property Type**, sélectionnez **Repository** et cliquez sur le bouton **[...]** afin de sélectionner le service, la granularité du port et l'opération.

- Cliquez sur **OK**.
- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin de voir le schéma.

- Cliquez sur **OK**.
- Connectez le **tESBProviderRequest** au **tLogRow_1**.
- Double-cliquez sur le **tLogRow_1** afin d'afficher sa vue **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et configurez le schéma comme suit.

- Reliez le **tLogRow_1** au **tXMLMap**.
- Reliez le **tXMLMap** au **tLogRow_2** et nommez la connexion *payload*.
- Dans l'espace de modélisation graphique, double-cliquez sur le **tXMLMap** afin d'ouvrir le **Map Editor**.
- Dans la partie inférieure droite de l'éditeur, cliquez sur le bouton **[+]** pour ajouter une ligne à la table *payload*. Nommez cette ligne *payload*.
- Dans la colonne **Type** de la ligne *payload*, sélectionnez le type de données **Document**. La racine XML correspondante est ajoutée automatiquement à la table supérieure de droite, représentant le flux de sortie.
- Dans la table *payload*, cliquez-droit sur *root* pour ouvrir le même menu contextuel.

- Dans le menu contextuel, sélectionnez **Create Sub-Element** et saisissez *response* dans la boîte de dialogue.
- Cliquez-droit sur le nœud *response* et sélectionnez **As loop element** dans le menu contextuel.
- Répétez l'opération pour créer un sous-élément *request* du nœud *root* dans la table d'entrée et définissez le nœud *request* comme élément de boucle.
- Cliquez sur le nœud *request* dans la table d'entrée et déposez-le dans la colonne **Expression** dans la ligne du nœud *response* de la table de sortie.

- Cliquez sur **OK** pour valider le mapping et fermer l'éditeur
- Dans l'espace de modélisation graphique, double-cliquez sur le **tLogRow_2** afin d'afficher sa vue **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et configurez le schéma comme suit.

- Sauvegardez le Job fournisseur.

Le Job consommateur comprend un **tFixedFlowInput**, un **tXMLMap**, un **tESBConsumer** et deux **tLogRow**.

- Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tFixedFlowInput**, un **tXMLMap**, un **tESBConsumer** et deux **tLogRow**.

- Double-cliquez sur le composant **tFixedFlowInput** afin d'afficher sa vue **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'éditer le schéma.

- Cliquez sur le bouton **[+]** pour ajouter une nouvelle ligne de type **String** et nommez-la *payloadString*.
- Cliquez sur **OK**.
- Dans le champ **Number of rows**, définissez le nombre de lignes à *1*.
- Dans la zone **Mode**, sélectionnez **Use Single Table** et saisissez `world` entre guillemets, dans le champ **Value**.
- Reliez le **tFixedFlowInput** au **tXMLMap**.
- Connectez le **tXMLMap** au **tESBConsumer** et nommez la connexion *payload*.
- Dans l'espace de modélisation graphique, double-cliquez sur le **tXMLMap** pour ouvrir le **Map Editor**.
- Dans la table de sortie, cliquez-droit sur *root* pour ouvrir son menu contextuel.
- Dans le menu contextuel, sélectionnez **Create Sub-Element** et saisissez *request* dans la boîte de dialogue.
- Cliquez-droit sur le nœud *request* et sélectionnez **As loop element** dans le menu contextuel.
- Cliquez sur le nœud *payloadstring* de la table d'entrée et déposez-le dans la colonne **Expression** de la ligne du nœud *request* dans la table de sortie.

- Cliquez sur **OK** pour valider le mapping et fermer le **Map Editor**.

- Démarrez le Job fournisseur. Vous pouvez voir, dans le log d'exécution :

```
...
 web service [endpoint: http://127.0.0.1:8088/esb/.provider] published
...
```

- Dans la vue **Component** du **tESBConsumer**, configurez les propriétés du composant.

- Cliquez sur le bouton **[...]** à côté du champ **Service Configuration** pour ouvrir l'éditeur.

- Dans le champ **WSDL**, saisissez : *http://127.0.0.1:8088/esb/provider?WSDL*
- Cliquez sur le bouton **Refresh** afin de récupérer le nom du port et de l'opération.
- Cliquez sur **OK**.

- Dans la vue **Basic settings** du composant **tESBConsumer**, configurez le schéma d'entrée (**Input Schema**) comme suit.

- Configurez le schéma de réponse (**Response Schema**) comme suit.

- Configurez le schéma (**Fault Schema**) comme suit.

- Connectez le **tESBConsumer** aux deux composants **tLogRow**.

- Dans l'espace de modélisation graphique, double-cliquez sur le composant **tLogRow_1** afin d'ouvrir sa vue **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et configurez le schéma comme suit.

- Dans l'espace de modélisation graphique, double-cliquez sur le **tLogRow_2** afin d'afficher sa vue **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et configurez le schéma comme suit.

- Sauvegardez le Job consommateur.
- Exécutez le Job fournisseur. Vous pouvez voir, dans le log d'exécution :

  2011-04-21 15:28:26.874:INFO::jetty-7.2.2.v20101205

  2011-04-21 15:28:27.108:INFO::Started

  SelectChannelConnector@127.0.0.1:8088

  web service [endpoint: http://127.0.0.1:8088/esb/provider] published

- Exécutez le Job consommateur. Vous pouvez voir, dans le log d'exécution du Job :

```
Starting job CallProvider at 14:15 21/04/2011.

[statistics] connecting to socket on port 3942
[statistics] connected
TEST_ESBProvider2
TEST_ESBProvider2SoapBingding
|
[tLogRow_2] payloadString: <request>world</request>
{http://talend.org/esb/service/job}TEST_ESBProvider2
```

```
{http://talend.org/esb/service/job}TEST_ESBProvider2SoapBinding
invoke
[tLogRow_1] payload: null
[statistics] disconnected
Job CallProvider2 ended at 14:16 21/04/2011. [exit code=0]
```

- Dans le log fournisseur, vous pouvez voir les traces :

```
web service [endpoint: http://127.0.0.1:8088/esb/provider]
published
[tLogRow_1] payload: <?xml version="1.0" encoding="UTF-8"?>
<request>world</request>
### world
[tLogRow_2] content: world
[tLogRow_3] payload: <?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://talend.org/esb/service/job">Hello, world!</response>
web service [endpoint: http://127.0.0.1:8088/esb/provider] unpublished
[statistics] disconnected
Job ProviderJob ended at 15:29 21/04/2011. [exit code=0]
```

tRESTClient

Propriétés du tRESTClient

Famille de composant	ESB/REST	
Fonction	Le composant **tRESTClient** envoie des requêtes HTTP et HTTPS à un fournisseur des services Web REST (REpresentational State Transfer) et obtient les réponses correspondantes. Ce composant s'intègre parfaitement dans *Talend Runtime* pour un support de HTTPS et, ultérieurement, le support de fonctionnalités de qualité de service.	
Objectif	Le composant **tRESTClient** est utilisé pour interagir avec des fournisseurs de services Web RESTful, en envoyant des requêtes HTTP et HTTPS, à l'aide de CXF (JAX-RS).	
Basic settings	*URL*	Saisissez l'URL du serveur REST à invoquer. Lorsque la case **Use Service Locator** est cochée, ce champ ne s'affiche pas et l'URL du serveur REST est obtenue automatiquement du serveur du Service Locator.
	Relative Path	Saisissez le chemin d'accès relatif au serveur REST à invoquer. Par exemple, vous souhaitez accéder à `http://localhost:8888/services/Customers/list` : Si la case **Use Service Locator** est décochée : vous pouvez saisir la première partie de l'adresse dans le champ **URL** et la seconde partie dans le champ **Relative Path**. Par exemple, saisissez `http://localhost:8888` dans le champ **URL** et `/services/Customers/list` dans le champ **Relative Path**. Vous pouvez également saisir le chemin d'accès complet au serveur REST dans le champ **URL** et laisser vide le champ **Relative Path**. Si la case **Use Service Locator** est cochée : la partie dans le champ **URL** est donnée par le Service Locator. Dans ce cas, vous devez connaître la partie **URL** et spécifier la suite dans le champ **Relative Path**. Elle dépend du service interrogé. Par exemple, dans le **tRESTRequest**, vous pouvez paramétrer l'endpoint **REST Endpoint** à `http://localhost:8888/services` et cocher la case **Use Service Locator**. Si vous souhaitez utiliser ce service, du côté **tRESTClient**, vous devez spécifier **/customers/list** dans le champ **Relative Path**.
	HTTP Method	Dans cette liste, sélectionnez une méthode HTTP décrivant l'action souhaitée. Les significations des méthodes HTTP sont sujets aux définitions de votre fournisseur de service Web. Les définitions des méthodes HTTP généralement acceptées sont listées ci-dessous : **- GET** : récupère des données du côté serveur selon les paramètres donnés. **- POST** : charge des données dans le serveur selon les paramètres donnés. **- PUT** : met à jour des données selon les paramètres donnés, ou crée les données, si elles n'existent pas. **- DELETE** : supprime les données selon les paramètres donnés.
	Content Type	Sélectionnez **XML**, **JSON** ou **FORM** selon le type de média du contenu à charger dans le serveur. Cette liste apparaît lorsque vous sélectionnez la méthode HTTP **POST** ou **PUT**.

	Accept Type	Sélectionnez le type de média que le côté client est préparé à accepter pour la réponse, du côté serveur. Les options disponibles sont : **XML**, **JSON** et **ANY**. Lorsque **ANY** est sélectionnée, le message de réponse peut être de tout type et sera transformé en une chaîne de caractères (string).
	Query parameters	Spécifiez les paramètres de la requête sous forme de paires nom-valeur. Cette option est généralement utilisée avec la méthode GET.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Ce composant utilise trois schémas built-in, en lecture seule. Cliquez sur **Edit Schema** pour visualiser la structure du schéma. *Modifier le type du schéma peut conduire à une perte de cette structure et à un échec de l'exécution du composant.*
	Input Schema	Schéma des données d'entrée. Le schéma contient deux colonnes : - **body** : stocke le contenu des données d'entrée structurées - **string** : stocke le contenu d'entrée lorsqu'il est de type string, ou est géré comme tel.
	Response Schema	Schéma pour la réponse du serveur. Ce schéma est passé au composant suivant à l'aide d'un lien **Row** > **Response** et contient trois colonnes: - **statusCode** : stocke le code de statut HTTP du serveur. - **body** : stocke le contenu d'une réponse structurée du serveur. - **string** : stocke le contenu de la réponse du serveur, lorsqu'elle est de type string ou gérée comme telle.
	Error Schema	Schéma pour les informations d'erreur. Ce schéma est passé au composant suivant via un lien **Row** > **Error** et contient deux colonnes : - **errorCode** : stocke le code de statut HTTP du serveur lorsqu'une erreur survient durant le processus d'invocation. L'interprétation des codes d'erreur spécifiques est soumise aux définitions de votre fournisseur de services Web. Pour des informations de référence, consultez le site http://fr.wikipedia.org/wiki/Liste_des_codes_HTTP. - **errorMessage** : stocke le message d'erreur correspondant au code d'erreur.
	Use Service Locator	Cochez cette case pour activer le Service Locator. Cela maintient la disponibilité du service et permet de répondre aux demandes et de respecter les (SLAs). Spécifiez l'espace de noms du Service ainsi que le nom du Service dans les champs correspondants.
	Use Service Activity Monitor	Cochez cette case pour activer le Service Activity Monitor. Il capture les événements et stocke les informations afin de faciliter les analyses en profondeur de l'activité des services et de suivre les messages via une transaction métier. Cette option peut être utilisée, entre autres, pour analyser le temps de réponse du service, identifier les modèles de trafic ou effectuer une analyse de cause racine.
	Use Authentication	Cochez cette case si l'authentification est requise du côté du serveur REST. Sélectionnez **Basic HTTP**, **SAML Token (ESB runtime only)** ou **OAuth2 Bearer** dans la liste. L'authentification **SAML Token** fonctionne uniquement avec le Runtime. Si vous sélectionnez **Basic HTTP** ou **SAML Token (ESB Runtime only)**, vous devez saisir votre identifiant et votre mot de passe.

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Si vous utilisez **OAuth2 Bearer**, renseignez le champ **Bearer Token** avec votre jeton d'authentification encodé en base64.
	Use Business Correlation	Cochez cette case pour créer un ID de corrélation dans ce composant. Vous pouvez spécifier un ID de corrélation ID dans le champ **Correlation Value**. Dans ce cas, l'ID de corrélation est passé au service qu'il appelle, afin que les appels en chaîne des services soient groupés sous cet ID de corrélation. Si vous laissez ce champ vide, cette valeur est générée automatiquement lors de l'exécution. Lorsque cette option est activée, le **tRESTClient** extrait également l'ID de corrélation de l'en-tête de réponse et le stocke dans la variable du composant pour usage ultérieur dans le flux.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*Log messages*	Cochez cette case pour enregistrer l'échange de messages entre le service et le consommateur.
	Convert Response To DOM Document	Cochez cette case afin de convertir la réponse du serveur en type Document. Décochez cette case si vous souhaitez que la réponse soit traitée comme une chaîne de caractères.
	Drop JSON Request Root	Cette option apparaît lorsque vous avez sélectionné **POST** dans la liste **HTTP Method** et **JSON** dans la liste **Content Type**. Cochez cette case pour supprimer les éléments JSON racine.
	Wrap JSON Response	Cette option est disponible et activée par défaut lorsque l'option **JSON** est sélectionnée dans la liste **Accept Type** de la vue **Basic settings**. Lorsque cette case est cochée, la réponse est entourée d'un élément *root*. Décochez cette case si vous souhaitez supprimer cet élément *root* de la réponse.
	HTTP Headers	Saisissez la (les) paire(s) nom-valeur des en-têtes HTTP pour définir les paramètres de l'opération HTTP demandée. Pour les définitions spécifiques des en-têtes HTTP, consultez votre fournisseur de services Web REST. Pour des informations de référence, consultez en.wikipedia.org/wiki/List_of_HTTP_headers (en anglais).
	Disable chunked encoding	Cette option apparaît uniquement lorsque la méthode dans la liste **HTTP Method** est **POST** ou **PUT**. Cochez cette case pour désactiver l'encodage des payloads par morceaux.
	Service Locator Customer Properties	Cette table apparaît lorsque la case **Use Service Locator** est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ **Property Name** et **Property Value** respectivement.
	Service Activity Customer Properties	Cette table apparaît lorsque la case **Use Service Activity Monitor** est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ **Property Name** et **Property Value** respectivement.
	Connection timeout	Configurez le temps, en secondes, avant suspension, durant lequel le client attend pour établir une connexion. Si ce champ est configuré à *0*, le client continue indéfiniment à essayer d'ouvrir une connexion.

	Receive timeout	Configurez le temps, en secondes, durant lequel le client attend pour obtenir une réponse, avant suspension de la connexion. Si ce champ est configuré à *0*, le client attend indéfiniment.
	Use HTTP proxy	Cochez cette case si vous utilisez un serveur proxy. Une fois cochée, vous devez saisir les informations de connexion : hôte, port, identifiant et mot de passe.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table et renseignez le champ **Code** avec une variable de contexte, afin d'activer ou désactiver dynamiquement l'option **Use Authentication** ou **Use HTTP proxy** lors de l'exécution. Vous pouvez ajouter deux lignes à la table afin de configurer les deux options. Une fois qu'un paramètre dynamique est défini, l'option correspondante est mise en relief et devient inutilisable dans les vues **Basic settings** et **Advanced settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme client de service Web RESTful pour communiquer avec un fournisseur de services RESTful, avec la possibilité de saisir une requête dans un service, à l'intérieur d'un Job et de retourner le résultat du Job comme réponse du service. Selon les actions à effectuer, le composant peut être un composant de début ou intermédiaire.	
Connections	Liens de sortie : **Row** : Response ; Error. **Trigger** : OnSubjobOk ; OnSubjobError ; Run if ; OnComponentOk ; OnComponentError. Liens d'entrée : **Row** : Main ; Reject. **Trigger** : Run if ; OnSubjobOk ; OnSubjobError ; OnComponentOk ; OnComponentError. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **HEADERS** : en-têtes des réponses HTTP. Cette variable est une variable **Flow** et retourne une liste des valeurs des en-têtes des réponses HTTP. **CORRELATION_ID** : l'ID de corrélation par lequel regrouper les appels de services en chaîne. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Scénario 1 : Obtenir des informations sur un utilisateur en interagissant avec un service RESTful

Ce scénario décrit un Job composé de trois composants qui récupère des informations sur un utilisateur en se basant sur l'identifiant de l'utilisateur depuis un service REST via HTTP GET. Les informations récupérées, ainsi que l'échange de messages entre le client et le serveur, sont affichés dans la console **Run**.

Prérequis

Si vous êtes un utilisateur de la solution *Talend Open Studio for ESB*, créez un Job comme décrit dans *Scénario 2 : Utiliser les paramètres URI Query pour explorer les données d'une base de données*, exécutez le Job afin d'exposer un service REST et saisissez l'URL du service REST dans votre navigateur Web, `http://localhost:8088/users` dans cet exemple. Vous devriez voir des informations telles que les suivantes :

```
- <users>
  - <user id="1">
      <first_name>Ronald</first_name>
      <last_name>Johnson</last_name>
    </user>
  - <user id="2">
      <first_name>Theodore</first_name>
      <last_name>Harding</last_name>
    </user>
  - <user id="3">
      <first_name>Bill</first_name>
      <last_name>Roosevelt</last_name>
    </user>
  </users>
```

Si vous n'êtes pas un utilisateur de la solution *Talend Open Studio for ESB*, vous devez obtenir l'URL, la structure des données et les paramètres requis par le service REST appelé depuis votre fournisseur de service REST. Vous devez également apporter les modifications nécessaires aux configurations présentées dans le scénario.

Construire le Job

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique :
   - un **tRESTClient**, utilisé pour appeler le service REST et récupérer les informations sur l'utilisateur depuis le serveur,
   - un **tXMLMap**, utilisé pour adapter l'arborescence du service REST
   - et un **tLogRow** afin d'afficher les informations sur l'utilisateur récupérées dans la console **Run**.
2. Reliez le **tRESTClient** au **tXMLMap** à l'aide d'un lien **Row** > **Response**.
3. Reliez le premier **tXMLMap** au **tLogRow** à l'aide d'un lien **Row** > **Main**. Nommez cette connexion *out* dans cet exemple.
4. Reliez le second **tXMLMap** au **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main** et renommez ce lien *out*.

5. Afin de mieux identifier le rôle de chaque composant, renommez-les comme suit :

Configurer les composants

Configurer l'appel du service

1. Double-cliquez sur le **tRESTClient** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **URL**, saisissez l'URL du service REST invoqué, *"http://localhost:8088/users"* dans cet exemple. Notez que l'URL utilisée dans ce scénario est donnée uniquement à titre d'exemple.

3. Dans la liste **HTTP Method**, sélectionnez **GET** pour envoyer une requête HTTP afin de récupérer les enregistrements existants.

   Dans la liste **Accept Type**, sélectionnez le type accepté par le côté client pour la réponse du côté serveur, **XML**. Laissez les autres paramètres tels qu'ils sont.

4. Cliquez sur le bouton **[+]** situé sous la table **Query parameters** afin d'ajouter deux paramètres, *from* et *to*. Définissez la valeur de ces paramètres à *2* afin d'obtenir les informations de l'utilisateur ayant pour identifiant *2*.

Sinon, vous pouvez également obtenir les informations de l'utilisateur ayant pour identifiant *2* en ajoutant *?from=2&to=2* à l'URL du service.

5. Dans la vue **Advanced settings** du **tRESTClient**, cochez les cases **Log messages** et **Convert Response To DOM Document** afin d'enregistrer le contenu de l'échange de messages avec le serveur et convertir les réponses du serveur en type Document.

Mapper la structure du service et afficher les informations de l'utilisateur récupérées

1. Double-cliquez sur le **tXMLMap** pour ouvrir l'éditeur de mapping.

2. Si vous avez sélectionné **XML** dans la liste **Accept Type** du composant **tRESTClient**, définissez la structure de l'arbre XML selon la structure du service.

    1. Dans la table d'entrée, dans la colonne *body*, cliquez-droit sur le nœud *root*. Dans le menu contextuel, cliquez sur **Rename** et renommez le nœud *users*.

    2. Cliquez-droit sur le nœud *users*. Dans le menu contextuel, cliquez sur **Create Sub-Element** et créez un sous-élément nommé *users*. Définissez l'élément *user* en tant qu'élément de boucle car la structure XML du service Web à invoquer effectue une boucle sur cet élément.

    3. Cliquez-droit sur le nœud *user*. Dans le menu contextuel, cliquez sur **Create Attribute** et saisissez *id* dans la boîte de dialogue **[Create New Attribute]** afin de créer un attribut nommé *id* pour le nœud *user*.

    4. Cliquez-droit à nouveau sur le nœud *user*. Dans le menu contextuel, cliquez sur **Create Sub-Element** et saisissez *first_name* dans la boîte de dialogue **[Create New Element]** afin de créer un sous-élément nommé *first_name* pour le nœud *user*.

        Répétez cette opération afin de créer un autre sous-élément *last_name* sous le nœud *user*.

    5. Déposez les colonnes *id*, *first_name* et *last_name* de la table d'entrée dans la table de sortie. Cliquez ensuite sur **OK** pour valider le mapping et fermer l'éditeur.

Si vous avez sélectionné **JSON** dans la liste **Accept Type** du composant **tRESTClient**, la réponse du serveur est renvoyée au format JSON et convertie en type Document. Dans cet exemple, la structure de la réponse convertie se présente comme suit :

```
<root>
    <users>
        <user>
            <id>2</id>
            <first_name>Theodore</first_name>
            <last_name>Harding</last_name>
        </user>
    </users>
</root>
```

Notez que l'élément `<root>` est supprimé si la case **Wrap JSON Response** est cochée dans la vue **Advanced settings** du composant **tRESTClient**.

Définissez la structure XML en conséquence et mappez-la comme décrit précédemment.

3. Double-cliquez sur le **tLogRow** afin d'ouvrir sa vue **Basic settings**.

4. Cliquez sur le bouton **Sync columns** afin de vous assurer que le schéma du composant est synchronisé avec le schéma de sortie du **tXMLMap**.

5. Dans le champ **Mode**, sélectionnez l'option **Table** afin d'afficher le résultat de l'opération GET sous forme de tableau.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur la touche **F6** ou sur le bouton **Run** de la vue **Run** pour exécuter le Job.

   La console montre que le **tRESTClient** lit les informations sur l'utilisateur correspondant à l'identifiant spécifié depuis le serveur.

Si vous avez sélectionné **XML** dans la liste **Accept Type** du composant **tRESTClient**, les résultats d'exécution se présentent comme suit :

```
Execution
Run   Kill   Clear

[statistics] connecting to socket on port 3496
[statistics] connected
May 26, 2014 5:38:17 PM
org.apache.cxf.interceptor.LoggingOutInterceptor
INFO: Outbound Message
---------------------------
ID: 1
Address: http://localhost:8088/users?from=2&to=2
Http-Method: GET
Content-Type: */*
Headers: {Accept=[application/xml], Content-Type=[*/*]}
--------------------------------------
May 26, 2014 5:38:17 PM
org.apache.cxf.interceptor.LoggingInInterceptor
INFO: Inbound Message
----------------------------
ID: 1
Response-Code: 200
Encoding: ISO-8859-1
Content-Type: application/xml
Headers: {content-type=[application/xml], Date=[Mon, 26
May 2014 09:38:17 GMT], Server=
[Jetty(8.1.14.v20131031)], transfer-encoding=[chunked]}
Payload: <?xml version="1.0" encoding="UTF-8"?>
<users><user
id="2"><first_name>Theodore</first_name><last_name>Hard
ing</last_name></user></users>
--------------------------------------
.--+----------+--------.
|     Show User Info     |
|=-+----------+--------=|
|id|first_name|last_name|
|=-+----------+--------=|
|2 |Theodore  |Harding  |
'--+----------+--------'

[statistics] disconnected
Line limit 100   Wrap
```

Si vous avez sélectionné **JSON** dans la liste **Accept Type** du composant **tRESTClient**, les résultats d'exécution se présentent comme suit :

Scénario 2 : Mettre à jour les informations des utilisateurs via une interaction avec un service RESTful

Ce scénario décrit un Job à trois composants mettant à jour, dans une base de données distante, les informations d'une liste d'utilisateurs via un service REST, à l'aide de la méthode HTTP POST. Une fois exécuté, le Job affiche les informations de l'échange serveur-client dans la console de la vue **Run**.

Les informations à mettre à jour sur le serveur sont stockées dans un fichier CSV, qui se présente comme suit :

```
id;first_name;last_name
1;John;Smith
2;Martin;Reagan
3;James;White
4;Jenny;Jackson
5;Robert;Thomson
```

Prérequis:

Si vous êtes un utilisateur de la solution *Talend Open Studio for ESB*, créez un Job comme décrit dans *Scénario 3 : Service REST acceptant des requêtes HTTP POST* et exécutez le Job en tant que serveur REST afin d'exposer un service REST acceptant les requêtes HTTP POST. Une fois le Job exécuté, la console affiche les informations relatives à l'implémentation du service, notamment l'URL de l'endpoint du service, *http://localhost:8045/users* dans cet exemple. Si vous saisissez *http://localhost:8045/users?_wadl* dans votre navigateur Web, les informations de définition du service s'affichent comme ceci :

```
- <application>
    <grammars/>
  - <resources base="http://localhost:8045/users">
    - <resource path="/">
      - <resource path="post/{id}/{first_name}/{last_name}">
          <param name="last_name" style="template" type="xs:string"/>
          <param name="id" style="template" type="xs:int"/>
          <param name="first_name" style="template" type="xs:string"/>
        - <method name="POST">
            <request/>
          - <response>
              <representation mediaType="application/xml"/>
              <representation mediaType="text/xml"/>
              <representation mediaType="application/json"/>
            </response>
          </method>
        </resource>
      </resource>
    </resources>
  </application>
```

Si vous n'utilisez pas la solution *Talend Open Studio for ESB*, vous devez obtenir les informations relatives au service à partir de votre fournisseur de services REST, comme l'URL, le chemin d'accès à la ressource, ainsi que la structure des données. Vous devez également apporter au scénario les modifications nécessaires à vos configurations.

Construire le Job

1. Créez un Job et ajoutez les composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette**:
   - un **tFileInputDelimited**, pour lire les informations des utilisateurs à partir d'un fichier local,
   - un **tXMLMap**, pour adapter la structure d'entrée à la structure du service REST,
   - et un **tRESTClient**, utilisé pour appeler le service REST afin d'envoyer des données à la base de données distante.
2. Connectez le **tFileInputDelimited** au **tXMLMap** à l'aide d'un lien **Row** > **Main**.
3. Connectez le **tXMLMap** au **tRESTClient** à l'aide d'un lien **Row** > **Main** et renommez le flux de sortie, *request* dans cet exemple.
4. Renommez les composants afin mieux identifier leur rôle.

Configurer les composants

Configurer les données d'entrée et les mappings de structure

1. Double-cliquez sur le **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.

2. Spécifiez le fichier d'entrée dans le champ **File name**, renseignez le champ **Header** en saisissant *1* afin d'ignorer l'en-tête et laissez les autres paramètres tels qu'ils sont.
3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir la boîte de dialogue **[Schema]** et de modifier le schéma d'entrée comme suit. Ajoutez les colonnes suivantes :
   - *id*, de type **Integer**, de longueur (*Length*) *2*. Cochez la case **Key** pour en faire la colonne clé,

- *first_name*, de type **String**,
- *last_name*, de type **String**.

4. Double-cliquez sur le composant **tXMLMap** pour ouvrir son éditeur.
5. Renommez le nœud *root* dans la table de sortie : cliquez-droit sur le nœud, sélectionnez **Rename** dans le menu contextuel et spécifiez un nouveau nom dans la boîte de dialogue, *user* dans cet exemple.
6. Sélectionnez les trois colonnes de la table d'entrée et déposez-les sur le nœud *user*, puis sélectionnez l'option **Create as sub-element of target node** dans la boîte de dialogue afin de configurer ces colonnes comme des sous-éléments du nœud *user*. Cela fait, cliquez sur **OK** afin de valider les mappings et fermer l'éditeur.

Configurer l'appel du service

1. Double-cliquez sur le composant **tRESTClient** pour ouvrir sa vue **Basic settings**.

2. Renseignez le champ **URL** avec l'emplacement de l'URI où est accessible le service REST, *"http://localhost:8088/users"* dans cet exemple.

3. Dans le champ **Relative Path**, saisissez le chemin d'accès à la ressource, `"/post/" + row1.id+ "/" + row1.first_name + "/" + row1.last_name` dans cet exemple. Cela va envoyer les données de la ligne d'entrée au serveur via le chemin de la ressource.

4. Dans la liste **HTTP Method**, sélectionnez **GET** afin d'envoyer une requête HTTP pour récupérer les enregistrements existants.

   Dans la liste **Accept Type**, sélectionnez le type que le côté client accepte pour la réponse du côté serveur, **XML**.

5. Dans la vue **Advanced settings** du composant **tRESTClient**, cochez la case **Log messages** afin d'enregistrer les informations relatives à l'échange de messages avec le serveur.

   Laissez les autres paramètres tels qu'ils sont.

Exécuter le Job et vérifier les résultats

Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job et appuyez sur **F6** pour l'exécuter.

La console affiche les informations relatives à l'échange client serveur :

La console du Job utilisé en tant que serveur affiche les informations relatives à l'échange, ainsi que le résultat de la mise à jour de la base de données.

Execution

Run Kill Clear

```
ID: 5
Address: http://localhost:8045/users/post/5/Robert/Thomson
Encoding: ISO-8859-1
Http-Method: POST
Content-Type: application/xml
Headers: {Accept=[*/*], Cache-Control=[no-cache], connection=
[keep-alive], Content-Length=[123], content-type=
[application/xml], Host=[localhost:8045], Pragma=[no-cache],
User-Agent=[Apache CXF 2.7.11]}
Payload: <?xml version="1.0" encoding="UTF-8"?>
<user><id>5</id><first_name>Robert</first_name><last_name>Thomson<.
last_name></user>
--------------------------------------
.--+----------+---------.
|        tLogRow_1       |
|=-+----------+---------=|
|id|first_name|last_name|
|=-+----------+---------=|
|1 |John      |Smith    |
|2 |Martin    |Reagan   |
|3 |James     |White    |
|4 |Jenny     |Jackson  |
|5 |Robert    |Thomson  |
'--+----------+---------'

[INFO ]: org.apache.cxf.interceptor.LoggingOutInterceptor -
Outbound Message
---------------------------
ID: 5
Response-Code: 200
Content-Type: application/xml
Headers: {Content-Type=[application/xml], Date=[Thu, 29 May 2014
09:13:02 GMT]}
Payload: <?xml version="1.0" encoding="UTF-8"?>
<user><id>5</id><first_name>Robert</first_name><last_name>Thomson<.
last_name></user>
```

Line limit 100 Wrap

tRESTRequest

Propriétés du tRESTRequest

Ce composant est disponible pour les utilisateurs du Studio Talend avec option ESB.

Famille de composant	ESB/REST	
Fonction	Le **tRestRequest** est un composant côté serveur acceptant les requêtes HTTP et/ou HTTPS des clients et supportant les méthodes HTTP GET, POST, PUT et DELETE. Pour activer le support de HTTPS, vous devez générer un keystore et ajouter quelques propriétés liées à la configuration de sécurité HTTPS dans le fichier *org.ops4j.pax.web.cfg* de votre conteneur Talend Runtime. Pour plus d'informations, consultez le *Talend ESB Container Administration Guide* (en anglais).	
Objectif	Ce composant vous permet de recevoir les requêtes GET/POST/PUT/DELETE des clients, du côté serveur.	
Basic settings	*REST Endpoint*	Renseignez ce champ en saisissant l'emplacement de l'URI où le service Web RESTful sera accessible pour les requêtes. Si vous voulez que votre service soit disponible à la fois sur HTTP et HTTPS, renseignez ce champ avec un chemin relatif.
	REST API Mapping	Cliquez sur le bouton **[+]** sous la table de mapping pour ajouter des lignes pour spécifier des requêtes HTTP : **Output Flow** : Cliquez sur le bouton **[...]** afin de spécifier le nom d'un flux de sortie et configurez le schéma du flux de sortie dans la boîte de dialogue qui suit. Le schéma n'est pas obligatoire, donc, si vous ne devez pas passer des paramètres supplémentaires au composant **tRESTRequest**, vous pouvez laisser le schéma vide. Cependant, vous devez alimenter le schéma si vous avez des paramètres URI Path définis dans le champ **URI Pattern** ou si vous devez ajouter des paramètres facultatifs de requêtes, tels que URI Query, HTTP Header ou Form parameters à l'URI spécifiée dans le champ **REST Endpoint**. Ajouter un schéma dont le nom est *body* permet de récupérer le corps de la requête des méthodes POST ou PUT. Les types **Document**, **String** et **Byte[]** sont supportés. Si vous spécifiez des paramètres d'URI dans le schéma du flux de sortie, vous devez définir le type de paramètre dans le champ **Comment** du schéma. Par défaut, si vous laissez le champ **Comment** vide, le paramètre est considéré comme un paramètre Path. Voici une liste des valeurs supportées dans **Comment** : • vide ou *path* correspond au paramètre par défaut @PathParam, • *query* correspond à @QueryParam, • *form* correspond à @FormParam, • *header* correspond à @HeaderParam. • *matrix* correspond à @MatrixParam.

		• *multipart* correspond @Multipart de CXF, représentant le corps de la requête. Il peut être utilisé avec la méthode POST ou PUT HTTP. Il est recommandé de définir les valeurs par défaut de vos paramètres facultatifs (Header, Query, Form). pour ce faire, renseignez les colonnes **Default** du schéma. **HTTP Verb** : Sélectionnez une méthode HTTP (GET/POST/PUT/DELETE/OPTIONS/HEAD) dans la liste. **URI pattern** : Renseignez ce champ avec les URIs RESTful décrivant la ressource. **Consumes** : Sélectionnez dans la liste le type de format du contenu consommé que le composant utilise, entre **XML or JSON**, **XML**, **JSON**, **Form**, **Multipart** et **Any** lorsque la méthode HTTP est POST ou PUT. **Produces** : Lorsque la méthode HTTP est GET, POST, PUT ou DELETE, sélectionnez dans la liste le type de format du contenu produit que le composant utilise, entre **XML or JSON**, **XML**, **JSON**, **HTML** et **Any**, ou sélectionnez **<oneway>** dans la liste pour accepter les requêtes à sens unique.
	Use Service Locator	Cochez cette case pour activer le Service Locator. Cela maintient la disponibilité du service et permet de répondre aux demandes et de respecter les (SLAs). Spécifiez l'espace de noms du Service ainsi que le nom du Service dans les champs correspondants.
	Use Service Activity Monitor	Cochez cette case pour activer le Service Activity Monitor. Il capture les événements et stocke les informations afin de faciliter les analyses en profondeur de l'activité des services et de suivre les messages via une transaction métier. Cette option peut être utilisée, entre autres, pour analyser le temps de réponse du service, identifier les modèles de trafic ou effectuer une analyse de cause racine.
	Use Authentication (ESB runtime only)	Cochez cette case pour activer l'option d'authentification dans ce service. Sélectionnez **Basic HTTP** ou **SAML Token**. Cette option fonctionne uniquement dans le Runtime.
	Use Business Correlation	Cochez cette case pour activer l'option de corrélation, afin que les appels en chaîne de services soient groupés sous le même ID de corrélation ID. Le **tRESTRequest** extrait l'ID de corrélation de l'en-tête de la requête et le stocke dans la variable du composant pour usage ultérieur dans le flux. Si cette option n'est pas activée du côté client, un ID de corrélation est automatiquement généré dans le **tRESTRequest**.
Advanced settings	*Log messages*	Cochez cette case pour enregistrer l'échange de messages entre le service et le consommateur.
	Wrap JSON Request	Cochez cette case pour entourer la requête JSON d'un élément *root*.
	Convert JSON values to String in response	Cochez cette case pour convertir les valeurs JSON en format String dans la réponse.
	Service Locator Customer Properties	Cette table apparaît lorsque la case **Use Service Locator** est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ **Property Name** et **Property Value** respectivement.
	Service Activity Customer Properties	Cette table apparaît lorsque la case **Use Service Activity Monitor** est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ **Property Name** et **Property Value** respectivement.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.	

	URI : URI de la requête REST. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **URI_BASE** : URI de base de la requête REST. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **URI_ABSOLUTE** : URI absolue de la requête REST. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **HTTP_METHOD** : méthode HTTP. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ATTACHMENT_HEADERS** : en-têtes des pièces jointes de la requête REST. Cette variable est une variable **Flow** et retourne une liste des valeurs des en-têtes des pièces jointes. **ATTACHMENT_FILENAMES** : nom des fichiers des pièces jointes de la requête REST. Cette variable est une variable **Flow** et retourne tous les noms de fichiers des pièces jointes. **PRINCIPAL_NAME** : nom du Principal de la requête REST. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CORRELATION_ID** : l'ID de corrélation par lequel regrouper les appels de services en chaîne. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant permet d'exposer un Job Talend en tant que service et de transmettre une requête à un service dans un Job puis retourne les résultats du Job en tant que réponse du service. Le composant **tRESTRequest** doit être utilisé avec le **tRESTResponse** afin de fournir les résultats d'un Job en tant que réponse, dans le cas d'une communication de style requête-réponse.

Scénario 1 : Service REST acceptant des requêtes HTTP GET et envoyant des réponses

Ce scénario décrit le processus d'acceptation d'une requête HTTP du client, son traitement et l'envoi de sa réponse.

Configurer le composant tRESTRequest

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tRESTRequest**, un **tXMLMap** et deux **tRESTResponse**.

2. Double-cliquez sur le composant **tRESTRequest** dans l'espace de modélisation graphique afin d'afficher sa vue **Basic settings**.

3. Dans le champ **REST Endpoint**, saisissez l'URI à laquelle le service Web REST sera accessible pour les requêtes, par exemple *"http://192.168.0.235:8088/user"*.

   Si vous voulez que votre service soit disponible à la fois sur HTTP et HTTPS, renseignez ce champ avec un chemin relatif. Par exemple, si vous saisissez *"/test"*, votre service sera disponible à la fois sur *http://<EndpointHTTPParDéfaut>/test* et *https://<EndpointHTTPSParDéfaut>/test*, à condition que vous ayez configuré votre conteneur d'exécution pour supporter l'HTTPS. Pour plus d'informations, consultez le *Talend ESB Container Administration Guide* (en anglais).

4. Cliquez sur le bouton **[+]** pour ajouter une ligne dans la table **REST API Mapping**.

5. Sélectionnez la nouvelle ligne et cliquez sur le bouton **[...]** de la colonne **Output Flow** afin d'ajouter un schéma pour le flux de sortie.

   Dans ce scénario, nommez le flux de sortie *GetOneUser*.

6. Cliquez sur le bouton **[+]** afin d'ajouter une ligne *id* au schéma. Cliquez sur **OK** pour sauvegarder le schéma.

7. Sélectionnez **GET** dans la colonne **HTTP Verb**.

8. Dans la colonne **URI Pattern**, saisissez `"/{id}/"`.

9. De la même manière, ajoutez une ligne dans la table **REST API Mapping** et nommez ce schéma *GetUserNumber*. Ajoutez une ligne string de type **String** au schéma. Saisissez *query* dans le champ **Comment**.

Sélectionnez **GET** dans la liste de la colonne **HTTP Verb**. Renseignez le champ dans la colonne **URI Pattern** en saisissant `"/number"`.

Configurer le tXMLMap

1. Reliez le composant **tRESTRequest** au **tXMLMap** à l'aide d'un lien **Row** > **GetOneUser**.

2. Double-cliquez sur le **tXMLMap** dans l'espace de modélisation graphique pour ouvrir le **Map Editor**.

3. Cliquez sur le bouton **[+]** en haut à droite afin d'ajouter une table de sortie et nommez-la *ResponseUsers*.

4. Cliquez sur le bouton **[+]** en bas à droite pour ajouter deux colonnes en sortie.

   Nommez la première colonne *body* et, dans la colonne **Type**, sélectionnez **Document**.

   Nommez la seconde colonne *string* et, dans la colonne **Type**, sélectionnez **String**.

5. Cliquez-droit sur le nœud **root** et sélectionnez **Create Sub-Element** pour créer un sous-élément. Nommez le sous-élément *foo* dans la boîte de dialogue qui s'ouvre.

6. Cliquez-droit sur le nœud *foo* et sélectionnez **As loop element**.

7. Sélectionnez la colonne *id* de la table *GetOneUser* et déposez-la dans le champ **Expression** du nœud *foo* dans la table *ResponseUsers*.

8. Cliquez sur **OK** afin de sauvegarder les paramètres.

Configurer le composant tRESTResponse

1. Reliez le **tXMLMap** à l'un des **tRESTResponse** à l'aide d'un lien **Row** > **ResponseUsers**.

   Le schéma défini dans le **tXMLMap** est automatiquement récupéré par le **tRESTResponse**.

2. Sélectionnez **OK(200)** dans la liste **Return status code**.

3. Laissez les autres paramètres tels qu'ils sont.

4. Reliez le **tRESTRequest** à l'autre **tRESTResponse** à l'aide d'un lien **Row** > **GetUserNumber**.

Sauvegarder et exécuter le Job

1. Sauvegardez le Job et appuyez sur **F6** pour l'exécuter.

```
[statistics] connecting to socket on port 3888
[statistics] connected
[INFO ]: org.apache.cxf.endpoint.ServerImpl - Setting the server's
publish address to be http://localhost:8088/user
[INFO ]: org.eclipse.jetty.server.Server - jetty-8.1.12.v20130726
[INFO ]: org.eclipse.jetty.server.AbstractConnector - Started
SelectChannelConnector@localhost:8088
```

2. Ouvrez votre navigateur si vous souhaitez tester le service.

La requête HTTP demandant l'ID d'un utilisateur est acceptée par le service REST et la réponse HTTP est retournée au serveur.

3. Répétez cette étape et saisissez *http://localhost:8088/user/number?string=123* dans la barre d'adresse. Appuyez sur la touche **Entrée**.

Une erreur de parsing XML est reportée, car le contenu du corps retourné est de type **String**. Cliquez-droit dans votre navigateur et sélectionnez **View Page Source** dans le menu contextuel menu.

Vous pouvez constater que *123* est retourné dans la réponse.

Scénario 2 : Utiliser les paramètres URI Query pour explorer les données d'une base de données

Ce scénario décrit comment utiliser les paramètres URI query dans le **tRESTRequest** afin d'explorer les données d'une base de données et d'envoyer la réponse via le **tRESTResponse**.

Pour ce faire, vous devez créer deux sous-jobs connectés à l'aide d'un lien **OnSubjobOk**. Ainsi, les deux sous-jobs seront exécutés séquentiellement. Pour plus d'informations concernant les connexions Trigger, consultez le *Guide utilisateur du Studio Talend*. Le premier sous-job crée et alimente la base de données. Le second permet d'explorer la base de données via le service REST.

Créer le premier sous-job

Pour ce faire, procédez comme suit :

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tFixedFlowInput** de la famille **Misc** et **tMysqlOutput** de la famille **Databases** > **Mysql**.

2. Reliez le **tFixedFlowInput** au **tMysqlOutput** à l'aide d'un lien **Row** > **Main**.

3. Double-cliquez sur le **tFixedFlowInput** pour afficher sa vue **Basic settings** :

4. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

5. Dans l'éditeur du schéma, cliquez trois fois sur le bouton **[+]** afin d'ajouter trois lignes. Paramétrez-les comme dans la capture d'écran.

6. Cliquez sur **OK**.

7. Retournez dans les propriétés du **tFixedFlowInput**, dans sa vue **Basic settings**. Dans la zone **Mode**, sélectionnez l'option **Use inline table**.

8. Sous le tableau, cliquez trois fois sur le bouton **[+]** afin d'ajouter trois lignes.

9. Dans le tableau, cliquez sur le champ *id* de la première ligne et saisissez *1*.

10. Cliquez sur le champ *firstname* de la première ligne, appuyez sur les touches **Ctrl+Espace** pour afficher la liste d'autocomplétion et sélectionnez la variable **TalendDataGenerator.getFirstName()** dans la liste.

11. Cliquez sur le champ *lastname* de la première ligne, appuyez sur les touches **Ctrl+Espace** pour afficher la liste d'autocomplétion et sélectionnez la variable **TalendDataGenerator.getLastName()** dans la liste.

12. Répétez l'opération pour les lignes suivantes, afin d'obtenir les mêmes paramètres que dans la capture d'écran.

13. Double-cliquez sur le **tMysqlOutput** pour afficher sa vue **Basic settings**.

14. Dans la liste **Property Type**, laissez sélectionné le mode **Built-in** et renseignez les champs **Host**, **Port**, **Database**, **Username** et **Password** manuellement. Si vous avez centralisé vos informations de connexion à

la base de données dans le nœud **Metadata** > **DB Connections** du **Repository**, vous pouvez sélectionner **Repository** dans la liste. Les champs seront automatiquement renseignés.

Pour plsu d'informations concernant le stockage des métadonnées, consultez le *Guide utilisateur du Studio Talend*.

15. Dans le champ **Table**, saisissez le nom de la table dans laquelle les données vont être chargées, par exemple : *users*.
16. Dans la liste **Action on table**, sélectionnez **Drop table if exists and create**, puis sélectionnez **Insert** dans la liste **Action on data**.
17. Cliquez sur **Sync columns** pour récupérer le schéma provenant du composant précédent.

Créer le second sous-job

Pour ce faire, procédez comme suit :

- Glissez-déposez les composants suivants, comme dans la première capture d'écran.
  - Un **tRESTRequest** et un **tRESTResponse** de la famille **ESB** > **REST**,
  - un **tFlowToIterate** de la famille **Orchestration**,
  - un **tMysqlInput** de la famille **Databases** > **Mysql**,
  - un **tXMLMap** de la famille **Processing**.

Configurer le composant tRESTRequest

Pour ce faire, procédez comme suit :

1. Double-cliquez sur le **tRESTRequest** dans l'espace de modélisation graphique afin d'afficher sa vue **Basic settings**.

2. Renseignez le champ **REST Endpoint** avec l'emplacement de l'URI où le service Web REST-ful sera accessible pour les requêtes. Par exemple, *"http://localhost:8088/users"*.

3. Cliquez sur le bouton **[+]** pour ajouter une ligne dans le tableau **REST API Mapping**.

4. Sélectionnez la nouvelle ligne et cliquez sur le bouton **[...]** dans la colonne **Output Flow** afin d'ajouter un schéma au flux de sortie.

5. Dans la boîte de dialogue, nommez le flux de sortie *getUsers*. Un éditeur de schéma s'ouvre.

6. Dans l'éditeur de schéma, cliquez deux fois sur le bouton **[+]** pour ajouter deux lignes et paramétrez-les comme dans la capture d'écran ci-dessus.

7. Cliquez sur **OK**.

8. Retournez dans les propriétés du **tRESTRequest**, dans sa vue **Basic settings**, sélectionnez **GET** dans la liste de la colonne **HTTP Verb**.

9. Laissez la colonne **URI Pattern** telle qu'elle est.

Vous avez créé le flux de sortie du **tRESTRequest**, vous pouvez donc utiliser le lien correspondant pour le connecter au composant suivant :

1. Connectez le **tRESTRequest** au **tFlowToIterate** à l'aide d'un lien **Row** > **getUsers**.

2. Laissez les paramètres du **tFlowToIterate** tels qu'ils sont.

3. Connectez le **tFlowToIterate** au **tMysqlInput** à l'aide d'un lien **Row** > **Iterate**.

Configurer le composant tMysqlInput

Pour ce faire, procédez comme suit :

1. Double-cliquez sur le **tMysqlInput** pour afficher sa vue **Basic settings**.

2. Dans la liste **Property Type**, sélectionnez **Built-in** et renseignez les champs **Host**, **Port**, **Database**, **Username** et **Password** manuellement. Si vous avez centralisé vos informations de connexion dans le nœud **Metadata** > **DB Connections** du **Repository**, vous pouvez sélectionner **Repository** dans la liste. Les champs seront automatiquement renseignés.

   Pour plus d'informations concernant le stockage des métadonnées, consultez le *Guide utilisateur du Studio Talend*.

3. Dans la liste **Schema**, sélectionnez **Built-in** et cliquez sur le bouton **[...]** à côté du champ **Edit schema**.

4. Dans l'éditeur de schéma, définissez le schéma exactement comme celui du **tFixedFlowInput**.

5. Dans le champ **Table Name**, renseignez le nom de la table dans laquelle les données sont stockées : *users*.

6. Dans la liste **Query Type**, sélectionnez **Built-in** et renseignez le champ **Query** avec la requête SQL suivante, permettant d'explorer les données de la base de données via la requête d'URI définie dans le composant **tRESTRequest**

```
"select * from users where id >= " + globalMap.get("getUsers.from") + " and  id <= "
 + globalMap.get("getUsers.to")
```

Configurer le composant tXMLMap

1. Cliquez-droit sur le composant **tMysqlInput**, maintenez et déposez sur le **tXMLMap** afin de connecter les deux composants.

2. Double-cliquez sur le **tXMLMap** dans l'espace de modélisation graphique pour ouvrir le **Map Editor**.

3. Cliquez sur le bouton **[+]** dans le coin supérieur droit afin d'ajouter une sortie. Nommez-la *ResponseUsers*.

4. Cliquez sur le bouton **[+]** en bas à droite pour ajouter deux colonnes de sortie.

   Nommez la première colonne *body* et paramétrez son **Type** à **Document**.

   Nommez la seconde colonne *string* et paramétrez son **Type** à **String**.

5. Cliquez-droit sur le nœud **root**, sélectionnez **Rename** dans la liste et renommez-le *users*

6. Cliquez-droit sur le nœud **users** et sélectionnez **Create Sub-Element** afin de créer un sous-élément. Nommez le sous-élément *user* dans la boîte de dialogue qui apparaît.

7. Cliquez-droit sur le nœud *user* créé à l'étape précédente et sélectionnez **As loop element**.

8. Sélectionnez la colonne *id* de la table *row2*, à gauche, et déposez-la sur le nœud *user* de la table *ResponseUsers*, à droite.

9. Dans la boîte de dialogue **[Selection]**, sélectionnez l'option **Create as attribute of target** puis cliquez sur **OK**.

10. Sélectionnez les colonnes *firstname* et *lastname* de la table *row2*et déposez-les sur le nœud *user* de la table *ResponseUsers*.

11. Dans la boîte de dialogue **[Selection]**, sélectionnez l'option **Create as sub-element of target node** et cliquez sur **OK**.

12. Cliquez sur l'icône de clé anglaise, en haut de la table *ResponseUsers* pour ouvrir le panneau de paramétrage.

13. Paramétrez la fonctionnalité **All in one** à **true**. Ainsi, toutes les données XML sont écrites dans un seul flux.

14. Cliquez sur **OK** afin de sauvegarder les paramètres.

Configurer le composant tRESTResponse

1. Connectez le **tXMLMap** au **tRESTResponse** à l'aide d'un lien **Row** > **ResponseUsers**.

2. Le schéma défini dans le **tXMLMap** est automatiquement récupéré par le **tRESTResponse**. Laissez les autres paramètres tels qu'ils sont.

Connecter les deux sous-jobs

Les deux sous-jobs sont créés, vous pouvez donc les connecter.

1. Cliquez-droit sur le composant **tFixedFlowInput** du premier sous-job.
2. Sélectionnez **Trigger** > **OnSubjobOk** dans la liste.
3. Cliquez sur le composant **tRESTRequest** du second sous-job.

Ainsi, lors de l'exécution du Job, le second sous-job est exécuté uniquement si l'exécution du premier s'est déroulée correctement.

Sauvegarder et exécuter le Job

1. Sauvegardez le Job et appuyez sur **F6** pour l'exécuter.

```
[statistics] connecting to socket on port 4020
[statistics] connected
[INFO ]: org.apache.cxf.endpoint.ServerImpl - Setting the server's publish
address to be http://localhost:8088/users
[INFO ]: org.eclipse.jetty.server.Server - jetty-8.1.14.v20131031
[INFO ]: org.eclipse.jetty.server.AbstractConnector - Started
SelectChannelConnector@localhost:8088
```

2. Ouvrez votre navigateur si vous souhaitez tester le service.

   Par exemple, utilisez la requête d'URI `?to=2` afin de récupérer les données des deux premiers utilisateurs.

La requête HTTP pour l'ID d'un utilisateur est acceptée par le service REST et la réponse HTTP est retournée au serveur.

Pour un cas d'utilisation appelant ce service REST à l'aide de composants *Talend Open Studio for ESB*, consultez *Scénario 1 : Obtenir des informations sur un utilisateur en interagissant avec un service RESTful*.

Scénario 3 : Service REST acceptant des requêtes HTTP POST

Ce scénario décrit un Job composé de deux sous-jobs : le premier sous-job expose un service REST acceptant des requêtes HTTP POST de clients REST, écrivant des données dans une base de données après réception d'une requête HTTP et affichant les informations de l'échange serveur-client dans la console de la vue **Run**. Le second sous-job affiche les résultats des mises à jour de la base de données.

Construire le Job

1. Créez un Job et ajoutez les composants suivants au Job en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette** : un **tRESTRequest**, un **tMysqlOutput**, un **tXMLMap** et un **tRESTResponse**.

2. Connectez le **tRESTRequest** au composant **tMysqlOutput** à l'aide d'un lien **Row** > **Main** et renommez le flux de sortie *request* dans cet exemple.

3. Reliez le composant **tMysqlOutput** au **tXMLMap** à l'aide d'un lien **Row** > **Main**.

4. Connectez le composant **tXMLMap** au **tRESTResponse** à l'aide d'un lien **Row** > **Main** que vous renommez *response*, par exemple. Cliquez sur **OK** dans la boîte de dialogue qui s'ouvre afin d'accepter la propagation du schéma du **tRESTResponse**.

5. Ajoutez un **tMysqlInput** et un **tLogRow** au Job en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette**. Reliez le **tMysqlInput** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

6. Connectez le **tMysqlOutput** au **tMysqlInput** à l'aide d'un lien **Trigger** > **OnComponentOk** afin de relier les deux sous-jobs.

Configurer les composants

Configurer les paramètres de la requête REST

1. Double-cliquez sur le composant **tRESTRequest** pour ouvrir sa vue **Basic settings**.

2. Renseignez le champ **REST Endpoint** en saisissant l'emplacement de l'URI où le service REST est accessible pour les requêtes, *"http://localhost:8045/users"* dans cet exemple.

3. Cliquez sur le nom du flux de sortie, *request* dans cet exemple, dans la colonne **Output Flow** de la table **REST API Mapping** pour afficher le bouton **[...]**. Cliquez sur ce bouton pour ouvrir la boîte de dialogue **[Schema]**. Définissez comme suit le schéma pour le flux *request*. Ajoutez trois colonnes :

   - *id*, de type **Integer**, de longueur (*Length*) *2*. Cochez la case **Key** pour en faire la colonne clé,
   - *first_name*, de type **String**,
   - *last_name*, de type **String**.

   Cela fait, cliquez sur **OK** afin de valider les paramètres du schéma et fermer la boîte de dialogue.

4. Cliquez sur la colonne **HTTP Verb** et sélectionnez **POST** dans la liste.

5. Dans la colonne **URI Pattern**, saisissez le modèle d'URI des requêtes POST, `"/post/{id}/{first_name}/{last_name}"`, dans cet exemple.

   Laissez les autres paramètres tels qu'ils sont.

6. Cliquez sur l'onglet **Advanced settings** et cochez la case **Log messages**.

Configurer les paramètres de la base de données et de la réponse

1. Double-cliquez sur le composant **tMysqlOutput** pour ouvrir sa vue **Basic settings**.

2. Dans la liste **Property Type**, laissez **Built-in** sélectionné et renseignez manuellement les détails de connexion à la base de données dans les champs **Host**, **Port**, **Database**, **Username** et **Password**.

   Si vous avez centralisé vos informations de connexion à la base de données dans le nœud **Metadata** > **DB Connections** du **Repository**, vous pouvez choisir **Repository** dans la liste **Property Type** et sélectionner la connexion centralisée afin de renseigner automatiquement les champs. Pour plus d'informations concernant le stockage des métadonnées, consultez le *Guide utilisateur du Studio Talend*.

3. Renseignez le champ **Table** avec le nom de la table cible de la base de données, *users* dans cet exemple.

4. Sélectionnez les actions à effectuer sur la table de la base de données et sur les données selon vos besoins, dans les listes **Action on table** et **Action on data**, respectivement. Dans cet exemple, la table cible est créée si elle n'existe pas dans la base de données spécifiée. Les données provenant des requêtes clients sont insérées, ou mises à jour si elles existent déjà, dans la table de la base de données.

5. Cliquez sur le bouton **Sync columns** afin de synchroniser le schéma de la table et le schéma d'entrée.

6. Double-cliquez sur le composant **tXMLMap** pour ouvrir son éditeur de mapping.

7. Renommez le nœud *root* dans la table de sortie : cliquez-droit sur le nœud, sélectionnez **Rename** dans le menu contextuel et spécifiez un nouveau nom dans la boîte de dialogue qui s'ouvre, *user* dans cet exemple.

8. Sélectionnez les trois colonnes de la table d'entrée et déposez-les sur le nœud *user*, puis sélectionnez l'option **Create as sub-element of target node** dans la boîte de dialogue afin de configurer ces colonnes comme des sous-éléments du nœud *user*. Cela fait, cliquez sur **OK** afin de valider les mappings et fermer l'éditeur.

9. Double-cliquez sur le composant **tRESTResponse** et configurez les paramètres de réponse selon vos besoins. Dans cet exemple, laissez les paramètres par défaut.

Configurer l'affichage des résultats

1. Double-cliquez sur le **tMysqlInput** pour ouvrir sa vue **Basic settings**.

2. Spécifiez les informations de connexion à la base de données, le nom de la table ainsi que le schéma de la table. Ces informations sont les mêmes que dans le composant **tMysqlOutput**.

3. Cliquez sur le bouton **Guess Query** afin de renseigner le champ **Query** avec l'instruction de requête.

4. Double-cliquez sur le **tLogRow** et sélectionnez l'option **Table** afin d'afficher le contenu de la base de données sous forme de tableau.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Cliquez sur le bouton **Run** dans l'onglet **Run** ou appuyez sur la touche **F6** pour l'exécuter.

   La console affiche les informations relatives à l'implémentation du service, notamment son URL.

   Lorsqu'une requête HTTP POST est reçue d'un client, la console affiche les informations relatives à l'échange, ainsi que le résultat de la mise à jour de la base de données.

```
Execution
Run   Kill   Clear
SelectChannelConnector@localhost:8045
[INFO ]: org.apache.cxf.interceptor.LoggingInInterceptor - Inbound Message
----------------------------
ID: 1
Address: http://localhost:8045/users/post/4/John/Smith
Encoding: ISO-8859-1
Http-Method: POST
Content-Type: application/xml
Headers: {Accept=[application/xml], Cache-Control=[no-cache], connection=
[keep-alive], Content-Length=[127], content-type=[application/xml], Host=
[localhost:8045], Pragma=[no-cache], User-Agent=[Apache CXF 2.7.11]}
Payload: <?xml version="1.0" encoding="UTF-8"?>
<customer><id>4</id><first_name>John</first_name><last_name>Smith</last_name>
</customer>
--------------------------------------
.--+----------+---------.
|       tLogRow_1        |
|=-+----------+--------=|
|id|first_name|last_name|
|=-+----------+--------=|
|1 |Jimmy     |McKinley |
|2 |Martin    |Reagan   |
|3 |George    |Lundon   |
|4 |John      |Smith    |
'--+----------+---------'
[INFO ]: org.apache.cxf.interceptor.LoggingOutInterceptor - Outbound Message
----------------------------
ID: 1
Response-Code: 200
Content-Type: application/xml
Headers: {Content-Type=[application/xml], Date=[Wed, 28 May 2014 08:22:13
GMT]}
Payload: <?xml version="1.0" encoding="UTF-8"?>
<user><id>4</id><first_name>John</first_name><last_name>Smith</last_name></u
ser>
--------------------------------------
```

Pour un cas d'utilisation appelant un service REST à l'aide de la méthode HTTP POST, consultez *Scénario 2 : Mettre à jour les informations des utilisateurs via une interaction avec un service RESTful*.

tRESTResponse

Propriétés du tRESTResponse

Ce composant est disponible pour les utilisateurs du Studio Talend avec option ESB.

Famille de composant	ESB/REST	
Fonction	Le composant **tRestResponse** envoie une réponse HTTP et/ou HTTPS au client à la réception d'une requête HTTP et/ou HTTPS. Pour activer le support de HTTPS, vous devez générer un keystore et ajouter quelques propriétés liées à la configuration de sécurité HTTPS dans le fichier *org.ops4j.pax.web.cfg* de votre conteneur Talend Runtime. Pour plus d'informations, consultez le *Talend ESB Container Administration Guide* (en anglais).	
Objectif	Ce composant vous permet de retourner un code de statut HTTP spécifique au client comme réponse à une requête HTTP.	
Basic settings	*Return Body Type*	Sélectionnez dans la liste le type du corps de la réponse, entre **Document**, **String** et **Byte[]**.
	Return status code	Sélectionnez un code de statut dans la liste pour indiquer le statut de la requête. **- <<Custom>>** : Cette option vous permet de personnaliser le code de statut. Saisissez dans le champ le code de statut de votre choix. **- Bad Request (400)** : La syntaxe de la requête est mauvaise ou la requête est impossible à effectuer. **- Internal Server Error (500)** : Des conditions inattendues rencontrées par le serveur l'empêchent d'exécuter la requête. **- OK (200)** : La requête a été exécutée. **- Resource Not Found (404)** : Le serveur ne trouve aucune correspondance avec l'URI donnée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est en mode **Built-in**, c'est-à-dire créé et stocké localement pour ce composant. Cliquez sur **Edit schema** pour modifier le schéma. Le flux d'entrée est récupéré du composant connecté au **tRESTResponse**. Toute modification dans le flux d'entrée du **tRESTResponse** est automatiquement propagée au composant précédent. Le flux de sortie du **tRESTResponse** est un champ codé en dur nommé **body**, dont le type est défini dans la liste **Return Body Type**.
Advanced settings	*Response Headers*	Saisissez la(les) paire(s) nom-valeur pour passer des réponses supplémentaires concernant la réponse. Pour plus d'informations concernant les en-têtes de réponses, consultez le site http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html#sec6.2 (en anglais).
	Drop JSON Response Root	Cochez cette case pour déposer l'élément JSON racine dans la réponse.

	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant permet d'exposer un Job **Talend** en tant que service et de transmettre une requête à un service dans un Job puis retourne les résultats du Job en tant que réponse du service. Le composant **tRESTResponse** doit être utilisé avec le **tRESTRequest** afin de fournir les résultats d'un Job en tant que réponse, dans le cas d'une communication de style requête-réponse.	

Scénario associé

Pour un scénario dans lequel le **tRESTResponse** est utilisé, consultez *Scénario 1 : Service REST acceptant des requêtes HTTP GET et envoyant des réponses*.

tRouteFault

Propriétés du tRouteFault

Ce composant est disponible pour les utilisateurs du Studio Talend avec option ESB.

Famille de composant	ESB/Route	
Fonction	Le composant **tRouteFault** envoie des messages d'un Job à une Route et marque les messages comme Fault.	
Objectif	Le **tRouteFault** vous permet d'échanger des messages entre un Job d'intégration de données et une Route de médiation.	
Basic settings	*Output Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** afin de récupérer automatiquement les colonnes du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé.
	Camel	Saisissez les paramètres de message correspondant aux colonnes définies dans le schéma via le bouton **Edit schema**.
	Type	Choisissez le type : **Body** : Utilisé pour configurer le corps d'une Route, équivaut au code `exchange.getOut().setBody(...)`. **Property** : Utilisé pour configurer la propriété d'une Route, équivaut au code `exchange.setProperty(name, value)`. **System** : Utiliser pour configurer la propriété du système, équivaut au code `System.setProperty(name, value)`. **Header** : Utilisé pour configurer l'en-tête de la sortie, équivaut au code `exchange.getOut().setHeader(name, value)`.
	Name	Cette colonne est utilisée pour configurer le nom du paramètre lorsque son type est **Property**, **System** ou **Header**. Cette colonne est ignorée lorsque son type est **Body**.

Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de fin et nécessite un composant d'entrée.	
Limitation	Un Job utilisant un **tRouteFault** doit être lancé dans une Route comprenant le composant **cTalendJob**.	

Scénario : Récupérer des messages du tRouteFault

Ce scénario décrit comment envoyer des messages d'un Job à une Route à l'aide du composant **tRouteFault**. Pour ce faire, vous devez construire un Job envoyant le message puis créer une Route contenant le composant **cTalendJob**, afin d'appeler le Job et de recevoir le message.

Pour plus d'informations concernant l'utilisation des composants de Médiation, consultez le *Guide de référence des Composants Mediation de Talend ESB*.

Créer un Job d'intégration de données

Dans cette section, vous allez créer un Job nommé *RouteCommunication* pour envoyer un message à une Route.

Déposer et relier les composants

1. Glissez-déposez un **tFixedFlowInput** et un **tRouteFault** de la **Palette** dans l'espace de modélisation graphique.
2. Cliquez-droit sur le composant **tFixedFlowInput**, sélectionnez **Row** > **Main** dans le menu contextuel menu puis cliquez sur le **tRouteFault**.

Configurer les composants

1. Double-cliquez sur le composant **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **Edit Schema**. Dans la boîte de dialogue du schéma, cliquez sur le bouton **[+]** afin d'ajouter une ligne de type **String** et nommez-la *body*. Cliquez sur **OK** pour fermer la boîte de dialogue.

3. Dans le champ **Number of rows**, configurez le nombre de lignes à *1*.

4. Dans la zone **Mode**, sélectionnez **Use Single Table** et saisissez la requête suivante entre guillemets doubles, dans le champ **Value** :

```
Hello from Talend Job!
```

5. Double-cliquez sur le composant **tRouteFault** pour afficher sa vue **Basic settings**.

6. Vérifiez que le **tRouteFault** comporte un élément *body*. Configurez son type à **Body**.

7. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

Créer une Route de Médiation

Dans cette section, passez en perspective **Mediation** et créez une Route recevant le message envoyé par le Job.

Déposer et relier les composants

1. Glissez-déposez un composant **cMessagingEndpoint**, un **cTry**, un **cTalendJob** et un **cProcessor** de la **Palette** dans l'espace de modélisation graphique.

2. Reliez les composants à l'aide de liens **Row** > **Route** comme dans la capture d'écran.

3. Renommez les composants afin de mieux identifier leur rôle au sein de la Route.

Configurer les composants

1. Double-cliquez sur le **cMessagingEndpoint** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **URI**, saisissez `"timer://myTimer?period=2000"` pour démarrer les échanges de messages toutes les 2000 millisecondes.

3. Double-cliquez sur le **cTalendJob** pour afficher sa vue **Basic settings**.

4. Sélectionnez le Job *RouteCommunication* dans le **Repository**.

5. Double-cliquez sur le composant **cProcessor** pour ouvrir sa vue **Basic settings**.

6. Personnalisez le code dans la zone **Code** pour afficher le corps du message et vérifier que le message est bien marqué comme Fault :

```
System.out.println(exchange.getIn());
System.out.print("Is message fault: ");
```

```
System.out.println(exchange.getIn().isFault());
```

7. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Route.

Exécuter la Route

Cliquez sur la vue **Run** pour l'afficher et puis sur le bouton **Run** afin de démarrer l'exécution de votre Route. Vous pouvez également appuyer sur **F6** pour l'exécuter. Dans le journal d'exécution, vous pouvez constater que le message apparaît.

tRouteInput

Propriétés du tRouteInput

Ce composant est disponible pour les utilisateurs du Studio Talend avec option ESB.

Famille de composant	ESB/Route	
Fonction	Le composant **tRouteInput** accepte des messages provenant d'une Route.	
Objectif	Le **tRouteInput** vous permet d'échanger des messages entre un Job d'intégration de données et une Route de médiation.	
Basic settings	*Input Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé.
	Camel	Saisissez les paramètres du message correspondant aux colonnes définies dans le schéma via le bouton **Edit schema** à l'aide du langage Simple Expression. Pour plus d'informations concernant le langage Simple Expression, consultez le site (en anglais) http://camel.apache.org/simple.html.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de début et requiert un composant de sortie.	

Limitation	Un Job utilisant un **tRouteInput** doit être lancé dans une Route utilisant le composant **cTalendJob**. Lorsque vous créez un Job d'intégration de données, vous devez utiliser un composant **tRouteInput** afin d'éviter que le Job référencé démarre automatiquement lorsqu'il est déployé dans **Talend Runtime**. Ainsi, le Job démarre lorsqu'il est appelé par la Route. Pour plus d'informations concernant l'utilisation du composant **cTalendJob**, consultez le *Guide de référence des Composants Mediation de Talend ESB*.

Scénario : Récupérer des messages d'une Route

Ce scénario décrit comment accepter des messages provenant d'une Route à l'aide du composant **tRouteInput**. Pour ce faire, vous devez construire un Job acceptant le message et une Route envoyant le message puis appelant le Job, à l'aide du composant **cTalendJob**.

Pour plus d'informations concernant l'utilisation des composants de Médiation, consultez le *Guide de référence des Composants Mediation de Talend ESB*.

Créer un Job d'intégration de données

Dans cette section, vous allez construire un Job nommé *RouteCommunication* acceptant le message d'une Route.

Déposer et relier les composants

1. Glissez-déposez un **tRouteInput** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Cliquez-droit sur le composant **tRouteInput**, sélectionnez **Row** > **Main** dans le menu contextuel puis cliquez sur le composant **tLogRow**.

Configurer les composants

1. Double-cliquez sur le **tRouteInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit Schema**. Dans la boîte de dialogue du schéma, cliquez sur le bouton **[+]** pour ajouter une ligne. Définissez son type à **String** et nommez-la *body*. Cliquez sur **OK** pour fermer la boîte de dialogue

3. Dans le champ **Simple Expression** de l'élément *body*, saisissez `"${in.body}"` pour obtenir le corps du message d'entrée provenant de la Route.

4. Le composant **tLogRow** permet de monitorer les échanges de messages et ne nécessite aucune configuration.

5. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.

Créer une Route de médiation

Dans cette section, passez en perspective **Mediation** et créez une Route envoyant le message au Job.

Déposer et relier les composants

1. Glissez-déposez un **cMessagingEndpoint**, un **cSetBody** et un **cTalendJob** de la **Palette** dans l'espace de modélisation graphique.

2. Reliez les composants à l'aide de liens **Row** > **Route** comme dans la capture d'écran.

3. Renommez les composants afin de mieux identifier leur rôle.

Configurer les composants

1. Double-cliquez sur le **cMessagingEndpoint** pour ouvrir **Basic settings**.

2. Dans le champ **URI**, saisissez `"timer://myTimer?period=2000"` pour démarrer l'échange de messages toutes les 2000 millisecondes.

3. Double-cliquez sur le composant **cSetBody** pour afficher sa vue **Basic settings**.

4. Sélectionnez **Simple** dans la liste **Language** et saisissez `Hello!` dans le champ **Expression**.

5. Double-cliquez sur le composant **cTalendJob** pour afficher sa vue **Basic settings**.

6. Sélectionnez le Job *RouteCommunication* du **Repository**.

7. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Route.

Exécuter la Route

Cliquez sur la vue **Run** pour l'afficher puis sur le bouton **Run** pour démarrer l'exécution de votre Route. Vous pouvez également appuyer sur **F6** pour l'exécuter. Dans le journal d'exécution, vous pouvez constater que le message apparaît.

tRouteOutput

Propriétés du tRouteOutput

Ce composant est disponible pour les utilisateurs du Studio Talend avec option ESB.

Famille de composant	ESB/Route	
Fonction	Le composant **tRouteOutput** envoie des messages d'un Job à une Route.	
Objectif	Le **tRouteOutput** vous permet d'échanger des messages entre un Job d'intégration de données et une Route de médiation.	
Basic settings	*Output Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** afin de récupérer automatiquement les colonnes du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé.
	Camel	Saisissez les paramètres de message correspondant aux colonnes définies dans le schéma via le bouton **Edit schema**.
	Type	Choisissez le type : **Body** : Utilisé pour configurer le corps d'une Route, équivaut au code `exchange.getOut().setBody(...)`. **Property** : Utilisé pour configurer la propriété d'une Route, équivaut au code `exchange.setProperty(name, value)`. **System** : Utiliser pour configurer la propriété du système, équivaut au code `System.setProperty(name, value)`. **Header** : Utilisé pour configurer l'en-tête de la sortie, équivaut au code `exchange.getOut().setHeader(name, value)`.
	Name	Cette colonne est utilisée pour configurer le nom du paramètre lorsque son type est **Property**, **System** ou **Header**. Cette colonne est ignorée lorsque son type est **Body**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé comme composant de fin et nécessite un composant d'entrée
Limitation	Un Job utilisant un **tRouteOutput** doit être lancé dans une Route comprenant un composant **cTalendJob**.

Scénario : Récupérer des messages du tRouteOutput

Ce scénario décrit comment envoyer des messages d'un Job à une Route, à l'aide du composant **tRouteOutput**. Pour ce faire, vous devez construire un Job pour envoyer le message puis créer une Route utilisant le composant **cTalendJob** afin d'appeler le Job et de recevoir le message.

Pour plus d'informations concernant l'utilisation des composants de Médiation, consultez le *Guide de référence des Composants Mediation de Talend ESB*.

Créer un Job d'intégration de données

Dans cette section, vous allez construire un Job nommé *RouteCommunication* envoyant un message à une Route.

Déposer et relier les composants

1. Glissez-déposez un composant **tFixedFlowInput** et un **tRouteOutput** de la **Palette** dans l'espace de modélisation graphique.

2. Cliquez-droit sur le **tFixedFlowInput**, sélectionnez **Row** > **Main** dans le menu contextuel puis cliquez sur le composant **tRouteOutput**.

Configurer les composants

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit Schema**. Dans la boîte de dialogue du schéma, cliquez sur le bouton **[+]** pour ajouter une ligne. Définissez son type à **String** et nommez-la *body*. Cliquez sur **OK** pour fermer la boîte de dialogue.

3. Dans le champ **Number of rows**, configurez le nombre de lignes à *1*.

4. Dans la zone **Mode**, sélectionnez **Use Single Table** et saisissez la requête suivante entre guillemets doubles, dans le champ **Value** :

```
Hello from Talend Job!
```

5. Double-cliquez sur le composant **tRouteOutput** pour afficher **Basic settings**.

6. Vérifiez que le **tRouteOutput** contient bien un élément *body*. Configurez son type à **Body**.

7. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.

Créer une Route de médiation

Dans cette section, vous allez une Route pour recevoir le message envoyé par le Job.

Déposer et relier les composants

1. Glissez-déposez un **cMessagingEndpoint**, un **cTalendJob** et un **cLog** de la **Palette** dans l'espace de modélisation graphique.

2. Reliez les composants à l'aide de liens **Row** > **Route** comme dans la capture d'écran.

3. Renommez les composants afin de mieux identifier leur rôle.

Configurer les composants

1. Double-cliquez sur le **cMessagingEndpoint** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **URI**, saisissez `"timer://myTimer?period=2000"` afin de démarrer les échanges de messages toutes les 2000 millisecondes.

3. Double-cliquez sur le composant **cTalendJob** pour afficher sa vue **Basic settings**.

4. Sélectionnez le Job *RouteCommunication* du **Repository**.

5. Le composant **cLog** permet de monitorer les échanges de messages et ne nécessite aucune configuration particulière.

6. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Route.

Exécuter la Route

Cliquez sur la vue **Run** pour l'afficher puis sur le bouton **Run** pour démarrer l'exécution de votre Route. Vous pouvez également appuyer sur la touche **F6** pour l'exécuter. Dans le journal d'exécution, vous pouvez voir l'échange de messages.

Job tRouteOutput
Basic Run
Debug Run
Advanced Settings
Target Exec
Execution
Run
Kill
Clear
[statistics] connecting to socket on port 3350
[statistics] connected
[timer://myTimer] Monitor_cLog_1
INFO Exchange[ExchangePattern:InOnly,
BodyType:String, Body:Hello from Talend Job!]
[timer://myTimer] Monitor_cLog_1
INFO Exchange[ExchangePattern:InOnly,
BodyType:String, Body:Hello from Talend Job!]
[timer://myTimer] Monitor_cLog_1
INFO Exchange[ExchangePattern:InOnly,
BodyType:String, Body:Hello from Talend Job!]
Line limit
100
Wrap

Composants File

Ce chapitre passe en revue les propriétés des principaux composants présents dans la famille **File** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **File** regroupe les composants qui lisent et écrivent des données dans tous types de fichiers à un seul ou plusieurs schémas, du plus courant au format le plus spécifique (dans les sous-familles **Input** et **Output**).

De plus, la sous-famille **Management** regroupe des composants dédiés aux fichiers qui peuvent effectuer diverses traitements sur les fichiers, notamment la décompression, la suppression, la copie, la comparaison de fichiers, etc.

tAdvancedFileOutputXML

tAdvancedFileOutputXML appartient à deux familles de composants : File et XML. Pour plus d'informations concernant le composant **tAdvancedFileOutputXML**, consultez *tAdvancedFileOutputXML*.

tApacheLogInput

Propriétés du tApacheLogInput

Famille de composant	File/Input	
Fonction	Le composant **tApacheLogInput** permet de lire le fichier de log d'accès d'un serveur Apache.	
Objectif	Le **tApacheLogInput** aide à la gestion effective du serveur Apache. Il est obligatoire lorsqu'il s'agit d'obtenir un retour sur l'activité et les performances du serveur, mais aussi sur les problèmes susceptibles de se produire.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Dans le contexte d'utilisation du **tApacheLogInput**, le schéma est en lecture seule.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	File Name	Chemin d'accès et nom du fichier, et/ou variable à traiter. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**.
Advanced settings	*Encoding*	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone. Il permet de créer un flux de données à l'aide d'un lien **Row** > **Main**, ainsi que de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données dont le type	

	ne correspond pas au type défini. Pour un exemple d'utilisation de ces deux liens, consultez *Scénario 2 : Extraire les données XML erronées dans un flux de rejet* du composant **tFileInputXML**.

Scénario : Lire un fichier de log d'accès Apache

Le scénario suivant crée un Job à deux composants permettant de lire le fichier de log d'accès d'un serveur Apache et d'afficher les données en sortie dans la console de log **Run**.

1. A partir de la **Palette**, cliquez-déposez les composants **tApacheLogInput** et **tLogRow** dans l'éditeur graphique.

2. Cliquez-droit sur le **tApacheLogInput** et sélectionnez une connexion **Row** > **Main** dans le menu contextuel. Cliquez ensuite sur le **tLogRow**, le lien approprié apparaît entre les deux composants.

3. Dans l'espace de modélisation graphique, sélectionnez le **tApacheLogInput**.

4. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du composant **tApacheLogInput**.

5. Configurer les champs **Property Type** et **Schema** en mode **Built-In**.

6. Cliquez sur le bouton **Edit schema** si vous souhaitez voir les colonnes (en mode read-only).

7. Dans le champ **File Name**, renseignez le chemin d'accès ou parcourez vos dossiers jusqu'au fichier de log d'accès que vous voulez lire.

8. Dans l'éditeur graphique, sélectionnez le composant **tLogRow** et cliquez sur la vue **Component** pour un définir la configuration de base (**Basic settings**). Pour plus d'informations, consultez *tLogRow*.

9. Appuyez sur **F6** pour exécuter le Job.

```
Starting job apach at 12:11 01/09/2008.
194.84.172.4|-|-|27/May/2007|03:06:12|+0200|GET|/forum/index.php|HTTP/
1.0|200|19024|http://www.talendforge.org/forum/index.php|Mozilla/4.0
(compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)
88.154.28.65|-|-|27/May/2007|05:44:21|+0200|GET|/bugs/view.php?id=205|
HTTP/1.0|200|19456|http://le-gall.net/pierrick/blog/|Mozilla/4.0
(compatible; MSIE 5.0; Windows 98)
72.9.105.42|-|-|27/May/2007|08:53:01|+0200|GET|/forum/register.php|HT
TP/1.0|200|21669|http://www.talendforge.org/register.php|Mozilla/4.0
(compatible; MSIE 7.0b; Windows NT 6.0)
76.171.68.99|-|-|27/May/2007|19:12:33|+0200|GET|/forum/login.php|HTTP/
1.0|200|17445|http://talendforge.org/login.php|Mozilla/4.0
(compatible; MSIE 6.0; Windows NT 5.1) Opera 7.54 [en]
Job apach ended at 12:11 01/09/2008. [exit code=0]
```

Les lignes de log du fichier défini sont affichées dans la console.

tCreateTemporaryFile

Propriétés du tCreateTemporaryFile

Famille du composant	File/Management	
Fonction	Le composant **tCreateTemporaryFile** permet de créer et d'organiser les fichiers temporaires.	
Objectif	Le **tCreateTemporaryFile** aide à la création d'un fichier temporaire et à sa redirection dans dossier défini. Ce composant permet également de conserver ou de supprimer ce fichier temporaire, selon le cas, après l'exécution du Job.	
Basic settings	*Remove file when execution is over*	Cochez cette case pour que le fichier temporaire soit supprimé après l'exécution du Job.
	Directory	Renseignez le chemin d'accès au dossier où sont stockés les fichiers temporaires.
	Template	Attribuez un nom au fichier temporaire en respectant le modèle.
	Suffix	Entrez une extension de fichier pour préciser le format que vous voulez donner au fichier temporaire.
Utilisation	Le composant **tCreateTemporaryFile** vous donne la possibilité de gérer vos fichiers temporaires pour que la mémoire soit libérée au profit d'autres projets, ce qui permet d'optimiser les performances système.	
Global Variables	**FILEPATH** : chemin où a été créé le fichier. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate. **Trigger** : Run if, OnSubjobOk, OnSubjobError, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Créer un fichier temporaire et y écrire des données

Le scénario suivant décrit un Job simple permettant de créer un fichier temporaire vide dans un dossier défini, puis d'y écrire des données et enfin de le supprimer après l'exécution du Job.

Déposer et relier les composants

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l'éditeur graphique : le **tCreateTemporaryFile**, le **tRowgenerator**, le **tFileOutputDelimited**, le **tFileInputDelimited** et le **tLogRow**.
2. Reliez le **tCreateTemporaryFile** au **tRowGenerator** à l'aide d'un lien de type **OnSubjobOk**.
3. Reliez le **tRowGenerator** au **tFileOutputDelimited** à l'aide d'un lien de type **Row Main**.
4. Reliez le **tRowGenerator** au **tFileInputDelimited** à l'aide d'un lien de type **OnSubjobOk**.
5. Reliez le **tFileInputDelimited** au **tLogRow** à l'aide d'un lien de type **Row Main**.

Configurer les composants

1. Dans l'éditeur graphique, sélectionnez le composant **tCreateTemporaryFile**.
2. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tCreateTemporaryFile**.

3. Cochez la case **Remove file when execution is over** pour supprimer, après l'exécution du Job, le fichier temporaire créé.

4. Cliquez sur le bouton **[...]** à côté du champ **Directory** pour parcourir vos dossiers jusqu'à celui où seront stockés les fichiers temporaires. Vous pouvez également entrer directement le chemin d'accès.

5. Dans le champ **Template**, renseignez le nom du fichier temporaire en veillant à ce qu'il respecte le format du modèle.

6. Dans le champ **Suffix**, renseignez l'extension du fichier pour indiquer le format que vous voulez donner au fichier temporaire.

7. Dans l'éditeur graphique, sélectionnez le **tRowGenerator** puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**).

8. Dans la liste déroulante **Schema**, choisissez le mode **Built-In**.

9. Cliquez sur le bouton **[...]** du champ **Edit schema** pour définir les données à passer au composant **tFileOutputDelimited**. Dans cet exemple, seules les données de la colonne *value* sont traitées.

10. Cliquez sur **OK** pour fermer la boîte de dialogue.

11. Cliquez sur le bouton **[...]** du champ **RowGenerator Editor** pour ouvrir la boîte de dialogue de l'éditeur.

12. Dans le champ **Number of Rows for Rowgenerator**, entrez le chiffre "5" pour générer cinq lignes puis cliquez sur **Ok** pour fermer la boîte de dialogue.

13. Dans l'éditeur graphique, sélectionnez le **tFileOutputDelimited** puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**).

14. Dans la liste déroulante **Property Type**, choisissez le mode **Built-In**.

15. Cliquez dans le champ **File Name** puis appuyez simultanément sur **Ctrl+Espace** pour accéder à la liste d'auto-complétion. Pour envoyer les données en sortie vers le fichier temporaire créé, sélectionnez l'option `tCreateTemporaryFile_1.FILEPATH` dans la liste des variables.

16. Renseignez les séparateurs de lignes et de champs dans leurs champs respectifs **Row Separator** et **Field Separator**.

17. Choisissez le mode **Built-In** dans la liste déroulante **Schema** puis cliquez sur **Sync columns** pour synchroniser les colonnes d'entrée et de sortie. Notez que la connexion **Row** permet d'alimenter automatiquement le schéma de sortie.

    Pour plus d'informations sur les types de schémas, consultez le *Guide utilisateur du Studio Talend*.

18. Dans l'éditeur graphique, sélectionnez le composant **tFileInputDelimited**.

19. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tFileInputDelimited**.

20. Choisissez le mode **Built-in** dans la liste déroulante **Property Type**.

21. Cliquez dans le champ **File Name** puis appuyez simultanément sur **Ctrl+Espace** pour accéder à la liste d'auto-complétion. Pour lire les données du fichier temporaire créé, sélectionnez `tCreateTemporaryFile_1.FILEPATH` dans la liste des variables.

22. Renseignez les séparateurs de lignes et de champs dans leurs champs respectifs **Row Separator** et **Field Separator**.

23. Dans la liste déroulante **Schema**, choisissez le mode **Built-In** puis cliquez sur le bouton **[...]** du champ **Edit schema** pour définir les données à passer au composant **tLogRow**. Dans cet exemple, le schéma est composé de la seule colonne *value*.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour enregistrer votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

```
Starting job tCreateTemporaryFile at 15:46 19/04/2010.

1
2
3
4
5
Job tCreateTemporaryFile ended at 15:46 19/04/2010. [exit code=0]
```

Le fichier temporaire est créé, lors de l'exécution du Job, dans le dossier qui a été défini et les cinq lignes générées sont inscrites dans ce fichier. Le fichier temporaire est supprimé après l'exécution du Job.

tChangeFileEncoding

Propriétés du tChangeFileEncoding

Famille de composant	File/Management	
Fonction	Le composant **tChangeFileEncoding** permet de modifier l'encodage d'un fichier donné.	
Objectif	Le **tChangeFileEncoding** transforme l'encodage des caractères d'un fichier donné, et génère un nouveau fichier en fonction de l'encodage des caractères transformés.	
Basic settings	*Use Custom Input Encoding*	Cochez cette case pour personnaliser le type d'encodage des caractères. Lorsque cette case est cochée, une liste des types d'encodage d'entrée apparaît, vous permettant soit de sélectionner un type d'encodage d'entrée parmi les encodages proposés, soit de spécifier vous-même un type d'encodage d'entrée en sélectionnant **CUSTOM**.
	Encoding	A partir de cette liste de types d'encodage, vous pouvez sélectionnez l'une des options proposées, ou personnaliser l'encodage des caractères en sélectionnant **CUSTOM**, et spécifier un type d'encodage de caractères.
	Input File Name	Chemin d'accès au fichier d'entrée.
	Output File Name	Chemin d'accès au fichier de sortie.
Utilisation	Ce composant doit être utilisé en standalone.	
Global Variables	**EXISTS** : résultat spécifiant si un fichier existe ou non. Cette variable est une variable **Flow** et retourne un booléen. **FILENAME** : nom du fichier traité. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario: Transformer l'encodage des caractères d'un fichier

Ce scénario Java décrit un Job très simple permettant de transformer l'encodage des caractères d'un fichier texte, et de générer un nouveau fichier en tenant compte du nouvel encodage des caractères.

1. A partir de la **Palette**, cliquez et déposez un composant **tChangeFileEncoding** dans l'espace de modélisation graphique.

2. Double-cliquez sur le composant **tChangeFileEncoding** pour afficher l'onglet **Basic settings** de sa vue **Component**.

3. Cochez la case **Use Custom Input Encoding**. Sélectionnez le type d'encodage "GB2312" dans la liste déroulante **Encoding**.

4. Dans le champ **Input File Name**, parcourez ou saisissez le chemin d'accès au fichier d'entrée.

5. Dans le champ **Output File Name**, parcourez ou saisissez le chemin d'accès au fichier de sortie.

6. Sélectionnez **CUSTOM** dans la liste déroulante **Encoding**, et saisissez "UTF-16" dans le champ texte.

7. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

Le type d'encodage du fichier *in.txt* est transformé, et *out.txt* est généré avec un type d'encodage UTF-16.

tFileArchive

Propriétés du tFileArchive

Famille de composant	File/Management	
Fonction	Le composant **tFileArchive** zippe un ou plusieurs fichier(s) en fonction des paramètres définis et place l'archive créée dans le répertoire souhaité.	
Objectif	Ce composant zippe un ou plusieurs fichier(s) pour un éventuel traitement.	
Basic settings	*Directory*	Saisissez le chemin d'accès au répertoire contenant les fichiers à zipper. **Subdirectories** : Cochez cette case si vous souhaitez zipper les fichiers des sous-dossiers du répertoire sélectionné.
	Archive file	Répertoire de destination et nom du fichier archive créé.
	Compress level	Choisissez le niveau de compression que vous souhaitez appliquer. **Best** : la qualité de compression sera optimale mais le temps de compression sera long. **Normal** : qualité et temps de compression intermédiaires. **Fast (no compression)** : la compression sera rapide mais la qualité de compression sera amoindrie.
	All files	Cochez cette case si vous souhaitez que tous les fichiers du dossier soient zippés. Décochez-la pour spécifier dans le tableau **Files** le ou les fichier(s) que vous désirez zipper. **Filemask** : saisissez un nom de fichier ou masque de fichier utilisant un caractère spécial ou une expression régulière.
	Create directory if not exists	Cochez cette case pour créer un nouveau dossier si votre dossier de destination n'existe pas.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Overwrite Existing Archive	Cette case est cochée par défaut. Elle vous permet de sauvegarder une archive en remplaçant l'archive existante. Cependant, si vous décochez la case, une erreur sera rapportée, le remplacement d'archive échouera et la nouvelle archive ne pourra être sauvegardée. Lorsque le remplacement échoue, le Job poursuit son exécution.
	Encrypt files	Cochez cette case si vous souhaitez que votre archive soit protégée par un mot de passe. Le champ **Enter Password** apparaît et vous permet de saisir votre mot de passe. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Lorsque cette option est utilisée, le fichier archive peut être dézippé uniquement par le composant **tFileUnarchive** et non par votre programme habituel. Pour plus d'informations concernant le **tFileUnarchive**, consultez *tFileUnarchive*.

	ZIP64 mode	Cette option permet de créer des archives avec l'extension .zip64. Trois modes sont disponibles : • ASNEEDED : les archives ayant une extension .zip64 sont automatiquement créées selon la taille du fichier. • ALWAYS : les archives ayant l'extension .zip64 sont créées, quelle que soit la taille du fichier. • NEVER : les archives ayant l'extension .zip64 ne sont pas créées, quelle que soit leur taille.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ARCHIVE_FILEPATH** : chemin vers le fichier d'archive. Cette variable est une variable **After** et retourne une chaîne de caractères. **ARCHIVE_FILENAME** : le nom du fichier archive. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main, Reject, Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate. **Trigger** : Run if, OnSubjobOk, OnSubjobError, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez la section relative aux types de connexion, dans le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être utilisé en standalone.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Zipper des fichiers à l'aide d'un tFileArchive

Ce scénario met en scène un Job formé d'un composant unique. Il a pour but de zipper des fichiers et de les enregistrer dans le répertoire choisi.

1. Cliquez-déposez le composant **tFileArchive** de la **Palette** dans l'espace de modélisation graphique.
2. Double-cliquez sur le composant afin d'afficher sa vue **Component**.

3. Dans le champ **Directory**, cliquez sur **[...]**, parcourez votre répertoire et sélectionnez le dossier ou le fichier que vous souhaitez compresser.

4. Cochez la case **Subdirectories** si vous désirez inclure dans l'archive les fichiers des sous-dossiers.

5. Renseignez ensuite le champ **Archive file**, en indiquant le répertoire de destination et le nom de votre archive.

6. Cochez la case **Create directory if not exists** si votre répertoire de destination n'existe pas encore et que vous souhaitez le créer.

7. Dans la liste **Compress level**, sélectionnez le niveau de compression que vous désirez appliquer à votre archive. Dans cet exemple, utilisez le niveau **normal**.

8. Décochez la case **All Files** si vous ne souhaitez compresser que des fichiers spécifiques.

9. Ajoutez une ligne dans le tableau à l'aide du bouton **[+]**, et cliquez sur le nom qui s'affiche. Saisissez entre étoiles une partie du nom du ou des fichier(s) que vous souhaitez compresser.

10. Appuyez sur **F6** afin d'exécuter votre Job.

Le **tArchiveFile** a compressé les fichiers sélectionnés et a créé le dossier dans le répertoire indiqué.

tFileCompare

Propriétés du tFileCompare

Famille de composant	File/Management	
Fonction	Le composant **tFileCompare** compare deux fichiers et fournit des informations de comparaison (en fonction d'un schéma en lecture seule).	
Objectif	Ce composant permet de contrôler la qualité des données des fichiers traités.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule.
	File to compare	Le chemin d'accès au fichier à vérifier.
	Reference file	Le chemin d'accès du fichier sur lequel la comparaison est basé.
	If differences are detected, display *If no difference detected, display*	Saisissez le message à afficher dans la console Run en fonction du résultat de la comparaison.
	Print to console	Cochez cette case pour afficher le message saisi.
Utilisation	Ce composant peut être utilisé en standalone mais il est généralement relié à un composant de sortie pour regrouper les informations de comparaison.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate. **Trigger** : Run if, OnSubjobOk, OnSubjobError, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.	
Global Variables	**DIFFERENCE** : résultat de la comparaison. Cette variable est une variable **Flow** et retourne un booléen. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Comparer des fichiers dézippés

Ce scénario décrit un Job permettant de dézipper un fichier et de le comparer avec un fichier de référence pour s'assurer qu'il n'a pas été modifié. La sortie de cette comparaison est stockée dans un fichier délimité et un message est affiché dans la console.

1. Cliquez et déposez les composants suivants : **tFileUnarchive**, **tFileCompare** et **tFileOutputDelimited**.
2. Reliez les composants **tFileUnarchive** et **tFileCompare** à l'aide d'une connexion de type **Iterate**.
3. Connectez le composant **tFileCompare** au composant de sortie à l'aide d'un lien **Main row**.
4. Dans l'onglet **Basic settings** du composant **tFileUnarchive**, renseignez le chemin d'accès au fichier à dézipper.
5. Dans le champ **Extraction Directory**, renseignez le chemin de destination du fichier dézippé.
6. Dans l'onglet Basic settings du composant **tFileCompare**, paramétrez les propriétés du fichier à comparer. Dans le champ **File to compare**, appuyez sur **Ctrl+Espace** pour afficher la liste des variables globales. Sélectionnez *$_globals{tFileUnarchive_1}{CURRENT_FILEPATH}* ou *"((String)globalMap.get("tFileUnarchive_1_CURRENT_FILEPATH"))"* en fonction du langage avec lequel vous travaillez, pour récupérer le chemin d'accès du fichier à partir du composant **tFileUnarchive**.

7. Puis dans le champ **Reference file**, paramétrez le fichier de référence sur lequel est basée la comparaison.
8. Dans les champs **If differences detected, display** et **If no differences detected, dispay**, paramétrez les messages à afficher lorsque les fichiers sont différents et lorsqu'ils sont identiques, par exemple :

```
"[Job " + JobName + "]
Files differ"
```

9. Cochez la case **Print to Console**, pour que le message défini apparaisse à la fin de l'exécution du Job.
10. Le schéma est en lecture seule et contient les informations standard. Cliquez sur **Edit schema** pour le consulter.

11. Paramétrez le composant de sortie en utilisant des points-virgules comme séparateurs de données.

12. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

```
Starting job CompareFiles at 14:11 19/06/2007.
[job CompareFiles] Files differ
Job CompareFiles ended at 14:11 19/06/2007. [exit code=0]
```

Le message paramétré apparaît dans la console et la sortie contient les informations telles qu'elles ont été définies dans le schéma.

```
file;file_ref;moment;job;component;differ;message
C:\Input\Accounts\Sunnyvale_accounts_new.xls;C:/Input/Sunnyvale_accounts.csv;
2007-06-19 14:11:59;CompareFiles;tFileCompare_1;1;[job CompareFiles] Files
differ
```

tFileCopy

Propriétés du tFileCopy

Famille de composant	File/Management	
Fonction	Le composant **tFileCopy** fait une copie d'un fichier source ou d'un dossier dans un répertoire cible.	
Objectif	Ce composant permet de rationaliser les traitements en effectuant des tâches récurrentes et fastidieuses de copie de manière automatique.	
Basic settings	*File Name*	Spécifiez le chemin d'accès au fichier à copier. Ce champ n'apparaît pas lorsque la case **Copy a directory** est cochée.
	Copy a directory	Cochez cette case pour copier un répertoire ainsi que tous les sous-répertoires et fichiers qu'il contient.
	Source directory	Spécifiez le répertoire source à copier. Ce champ apparaît uniquement lorsque la case **Copy a directory** est cochée.
	Destination directory	Spécifiez le répertoire dans lequel le fichier source ou le répertoire doit être copié.
	Rename	Cochez cette case pour renommer le fichier copié à la destination. Ce champ n'apparaît pas lorsque la case **Copy a directory** est cochée.
	Destination filename	Spécifiez un nouveau nom pour le fichier à copier. Ce champ apparaît uniquement lorsque la case **Rename** est cochée.
	Remove source file	Cochez cette case pour supprimer le fichier source après sa copie dans le répertoire de destination. Ce champ n'apparaît pas lorsque la case **Copy a directory** est cochée.
	Replace existing file	Cochez cette case pour écraser le fichier déjà existant. Ce champ n'apparaît pas lorsque la case **Copy a directory** est cochée.
	Create a directory if it doesn't exist	Cochez cette case pour créer le répertoire de destination spécifié s'il n'existe pas déjà. Ce champ n'apparaît pas lorsque la case **Copy a directory** est cochée.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Utilisation	Ce composant peut être utilisé en standalone.	
Global Variables	**DESTINATION_FILENAME** : nom du fichier cible. Cette variable est une variable **After** et retourne une chaîne de caractères. **DESTINATION_FILEPATH** : chemin d'accès au fichier cible. Cette variable est une variable **After** et retourne une chaîne de caractères. **SOURCE_DIRECTORY** : répertoire source. Cette variable est une variable **After** et retourne une chaîne de caractères.	

	DESTINATION_DIRECTORY : répertoire cible. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate. **Trigger** : Run if, OnSubjobOk, OnSubjobError, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Récupérer un fichier de la corbeille

Ce scénario décrit un Job faisant une boucle sur une liste de fichiers, copie chaque fichier du répertoire source vers le répertoire cible et supprime les fichiers copiés du répertoire source.

1. Créez un nouveau Job et ajoutez les composants **tFileList** et **tFileCopy** en saisissant leurs noms dans l'espace de modélisation graphique ou en les déposant de la **Palette**.

2. Reliez **tFileList** à **tFileCopy** à l'aide d'une connexion **Row** > **Iterate**.

3. Double-cliquez sur le composant **tFileList** pour ouvrir sa vue **Basic settings**.

4. Dans le champ **Directory**, parcourez votre système ou saisissez le répertoire sur lequel sera effectuée la boucle.

5. Double-cliquez sur le composant **tFileCopy** pour ouvrir sa vue **Basic settings**.

6. Dans le champ **File Name**, appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales et sélectionnez la variable *tFileList_1.CURRENT_FILEPATH* dans la liste pour remplir le champ avec *((String)globalMap.get("tFileList_1_CURRENT_FILEPATH"))*.

7. Dans le champ **Destination directory**, parcourez votre système ou saisissez le répertoire dans lequel doit être copié chaque fichier.

8. Cochez la case **Remove source file** pour supprimer les fichiers ayant été copiés.

9. Appuyez sur **Ctrl+S** pour enregistrer le Job et appuyez sur **F6** pour l'exécuter.

   Tous les fichiers dans le répertoire source spécifié sont copiés dans le répertoire de destination et supprimés du répertoire source.

tFileDelete

Propriétés du tFileDelete

Famille de composant	File/Management	
Fonction	Le composant **tFileDelete** supprime un fichier d'un répertoire spécifié.	
Objectif	Ce composant permet de rationaliser les traitements en effectuant des tâches récurrentes et fastidieuses de manière automatique.	
Basic settings	*File Name*	Chemin d'accès au fichier à supprimer. Ce champ n'apparaît pas si la case **Delete folder** ou la case **Delete file or folder** est cochée.
	Directory	Chemin d'accès au dossier à supprimer. Ce champ est disponible uniquement lorsque la case **Delete folder** est cochée.
	File or directory to delete	Saisissez le chemin d'accès au fichier ou au dossier que vous souhaitez supprimer. Ce champ est disponible uniquement lorsque la case **Delete file or folder** est cochée.
	Fail on error	Cochez cette case pour empêcher l'exécution du Job si une erreur survient, par exemple si le fichier à supprimer n'existe pas.
	Delete Folder	Cochez cette case pour afficher le champ **Directory**, dans lequel indiquer le chemin d'accès au dossier à supprimer.
	Delete file or folder	Cochez cette case pour afficher le champ **File or directory to delete**, dans lequel indiquer le chemin d'accès au dossier ou au fichier que vous souhaitez supprimer.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en standalone.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate. **Trigger** : Run if, OnSubjobOk, OnSubjobError, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez la section relative aux types de connexion, dans le *Guide utilisateur du Studio Talend*.	
Global Variables	**DELETE_PATH** : chemin du dossier ou fichier supprimé. Cette variable est une variable **After** et retourne une chaîne de caractères. **CURRENT_STATUS** : résultats d'exécution du composant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Supprimer des fichiers

Ce scénario décrit un Job très simple supprimant des fichiers d'un répertoire donné.

1. Cliquez et déposez les composants suivants : **tFileList**, **tFileDelete**, **tJava**.

2. Dans le champ **Directory** de l'onglet **Basic settings** du composant **tFileList**, spécifiez le répertoire sur lequel la boucle doit être effectuée.

3. Dans le champ **Filemask**, paramétrez le masque de fichier en *"*.txt"*. Ne cochez aucune case.

4. Dans l'onglet **Basic settings** du composant **tFileDelete**, paramétrez le champ **File Name** afin que le fichier sélectionné par le composant **tFileList** soit supprimé. Ceci permet de supprimer tous les fichiers d'un répertoire spécifié.

5. Appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales. La variable adéquate permettant de récupérer le fichier en cours est : `((String)globalMap.get("tFileList_1_CURRENT_FILEPATH"))`.

6. Puis dans le composant **tJava**, définissez le message à afficher dans la sortie standard (la console **Run**). Dans cet exemple, saisissez le script suivant dans le champ **Code** :

```
System.out.println( ((String)globalMap.get("tFileList_1_CURRENT_FILE"))

+ " has been deleted!" );
```

7. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

8. Appuyez sur **F6** our sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

```
Starting job FileDel at 18:29 20/06/2007.
16.txt has been deleted!
15.txt has been deleted!
14.txt has been deleted!
13.txt has been deleted!
12.txt has been deleted!
11.txt has been deleted!
10.txt has been deleted!
09.txt has been deleted!
08.txt has been deleted!
07.txt has been deleted!
06.txt has been deleted!
05.txt has been deleted!
04.txt has been deleted!
03.txt has been deleted!
02.txt has been deleted!
01.txt has been deleted!
Job FileDel ended at 18:29 20/06/2007. [exit code=0]
```

Le message défini dans le composant **tJava** apparaît dans la console pour chaque fichier supprimé à l'aide du composant **tFileDelete**.

tFileExist

Propriétés du tFileExist

Famille du composant	File/Management	
Fonction	Le composant **tFileExist** vérifie si un fichier existe ou non.	
Objectif	Le **tFileExist** aide à rationaliser les processus en automatisant les tâches récurrentes et fastidieuses, comme par exemple vérifier si un fichier existe ou non.	
Basic settings	*File Name*	Chemin d'accès au fichier dont vous voulez vérifier l'existence.
Utilisation	Ce composant peut être utilisé en standalone.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate. **Trigger** : Run if, OnSubjobOk, OnSubjobError, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez la section relatives aux types de connexions, dans le *Guide utilisateur du Studio Talend*.	
Global Variables	**EXISTS** : résultat spécifiant si un fichier existe ou non. Cette variable est une variable **Flow** et retourne un booléen. **FILENAME** : nom du fichier traité. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Vérifier l'existence d'un fichier et le créer s'il n'existe pas

Ce scénario décrit un Job simple qui permet de vérifier si le fichier spécifié existe, d'afficher un message à l'écran si le fichier n'existe pas, de lire les données d'un autre fichier d'entrée spécifié et de les écrire dans un fichier délimité en sortie.

Déposer et relier les composants

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l'éditeur graphique : le **tFileExist**, le **tFileInputDelimited**, le **tFileOutputDelimited** et le **tMsgBox**.

2. Reliez d'abord le **tFileExist** au **tFileInputDelimited** à l'aide d'un lien de type **OnSubjobOk**, puis au **tMsgBox** à l'aide d'un lien de type **Run If**.

3. Reliez le **tFileInputDelimited** au **tFileOutputDelimited** à l'aide d'un lien de type **Row Main**.

Configurer les composants

1. Dans l'éditeur graphique, sélectionnez le **tFileExist** puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**).

2. Dans le champ **File name**, renseignez le chemin d'accès au fichier ou parcourez vos dossiers à la recherche du fichier dont vous voulez vérifier l'existence.

3. Dans l'éditeur graphique, sélectionnez le composant **tFileInputDelimited** puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**).

4. Renseignez le champ **File Name** en parcourant vos dossiers jusqu'au fichier d'entrée que vous voulez lire.

   Si le chemin d'accès du fichier contient des caractères accentués, vous obtiendrez un message d'erreur lors de l'exécution du Job. Pour plus d'informations concernant la procédure à suivre lorsque les accents ne sont pas supportés, consultez le Guide d'installation et de migrationTalend.

5. Définissez les séparateurs de lignes et de champs dans leurs champs respectifs **Row separator** et **Field Separator**.

6. Définissez le nombre de lignes d'en-tête (**Header**), de pied de page (**Footer**) et le nombre maximal de lignes à traiter (**Limit**) en fonction de vos besoins. Dans ce scénario, notre tableau comporte une ligne d'en-tête.

7. Dans la liste déroulante **Schema**, choisissez le mode **Built-In**. Cliquez ensuite sur le bouton **[...]** du champ **Edit schema** pour définir les données à passer au composant **tFileOutputDelimited**. Définissez les données présentes dans le fichier à lire ("file2" dans ce scénario).

   Pour plus d'informations sur les types de schémas, consultez le *Guide utilisateur du Studio Talend*.

   Le schéma du fichier "file2" est composé de cinq colonnes : *Num*, *Ref*, *Price*, *Quant* et *Tax*.

8. Dans l'éditeur graphique, sélectionnez le composant **tFileOutputDelimited**.

9. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tFileOutputDelimited**.

10. Définissez le champ **Property Type** en mode **Built-in**.

11. Dans le champ **File name**, appuyez simultanément sur **Ctrl+Espace** pour accéder à la liste des variables et sélectionner FILENAME.

12. Définissez les séparateurs de lignes et de champs dans leurs champs respectifs **Row separator** et **Field Separator**.

13. Cochez la case **Include Header** car, dans cet exemple, le fichier "file2" comprend un en-tête.

14. Définissez le champ **Schema** en mode **Built-in** puis cliquez sur le bouton **Sync columns** pour synchroniser le schéma du fichier de sortie (file1) avec le schéma du fichier d'entrée (file2).

15. Dans l'éditeur graphique, sélectionnez le composant **tMsgBox**.

16. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tMsgBox**.

17. Cliquez sur le lien **If** pour en afficher les propriétés dans la vue **Basic settings**.

18. Dans le panneau **Condition**, appuyez simultanément sur **Ctrl+Espace** pour accéder à la liste de variable et sélectionner EXISTS. Entrez un point d'exclamation avant la variable pour en inverser le sens.

Enregistrer et exécuter le Job

1. Appuyez sur **Ctrl+S** pour enregistrer votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

Une boîte de dialogue apparaît pour confirmer que le fichier n'existe pas.

Cliquez sur **OK** pour fermer cette boîte de dialogue et continuer l'exécution du Job. Le fichier manquant, "file1" dans cet exemple, est écrit sous la forme d'un fichier délimité à l'endroit préalablement déterminé.

tFileInputARFF

Propriétés du tFileInputARFF

Famille de composant	File/Input	
Fonction	Le composant **tFileInputARFF** lit un fichier ARFF ligne à ligne et sépare des champs simples.	
Objectif	Ce composant ouvre un fichier et le lit ligne par ligne afin de le diviser en champs et d'envoyer ces champs au composant suivant, comme défini par le schéma, via une connexion Row.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant connexion de base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	File Name	Chemin d'accès et nom du fichier, et/ou variable à traiter. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*Encoding*	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

	tStatCatcher Statistics	Cochez cette case pour collecter les informations de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour lire un fichier et séparer les champs à l'aide du séparateur spécifié.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Afficher le contenu d'un fichier ARFF

Le scénario suivant décrit un Job formé de deux composants ayant pour objectif de lire ligne à ligne un fichier ARFF, de sélectionner des données délimitées et d'afficher la sortie dans la console de la vue **Run**.

Un fichier ARFF se présente comme suit :

```
@relation vote
@attribute 'handicapped-infants' { 'n', 'y'}
@attribute 'water-project-cost-sharing' { 'n', 'y'}
@attribute 'adoption-of-the-budget-resolution' { 'n', 'y'}
@attribute 'physician-fee-freeze' { 'n', 'y'}
@attribute 'el-salvador-aid' { 'n', 'y'}
@attribute 'religious-groups-in-schools' { 'n', 'y'}
@attribute 'anti-satellite-test-ban' { 'n', 'y'}
@attribute 'aid-to-nicaraguan-contras' { 'n', 'y'}
@attribute 'mx-missile' { 'n', 'y'}
@attribute 'immigration' { 'n', 'y'}
@attribute 'synfuels-corporation-cutback' { 'n', 'y'}
@attribute 'education-spending' { 'n', 'y'}
@attribute 'superfund-right-to-sue' { 'n', 'y'}
@attribute 'crime' { 'n', 'y'}
@attribute 'duty-free-exports' { 'n', 'y'}
@attribute 'export-administration-act-south-africa' { 'n', 'y'}
@attribute 'Class' { 'democrat', 'republican'}
@data
'n','y','n','y','y','y','n','n','n','y',?,'y','y','y','n','y','republican'
'n','y','n','y','y','y','n','n','n','n','n','y','y','y','n',?,'republican'
?,'y','y',?,'y','y','n','n','n','n','y','n','y','y','n','n','democrat'
```

Un fichier ARFF est généralement composé de deux parties : la description de la structure des données, correspondant aux lignes qui débutent par `@attribute`, et les données brutes, qui suivent l'expression `@data`.

Déposer et relier les composants

1. Glissez et déposez le composant **tFileInputARFF** de la **Palette** dans l'espace de modélisation graphique.

2. Glissez et déposez de la même manière le composant **tLogRow**

3. Cliquez-droit sur le composant **tFileInputARFF** et sélectionnez **Row > Main** dans le menu contextuel. Glissez ce lien vers le **tLogRow** et relâchez lorsque le symbole de prise de courant apparaît. La transmission du flux de donnée entre les deux composants est ainsi établie.

Configurer les composants

1. Double-cliquez sur le **tFileInputARFF** pour paramétrer ce composant.

2. Dans la vue **Component**, renseignez le champ **File Name**. Parcourez votre répertoire de fichiers, afin de sélectionner le fichier *.arff*.

3. Dans le champ **Schema**, sélectionnez **Built-In**.

4. Cliquez sur le bouton **[...]** du champ **Edit schema**, pour effectuer manuellement la description des colonnes du fichier lu.

5. Cliquez autant de fois sur le bouton que nécessaire pour décrire chacune des colonnes du fichier lu. Dans cet exemple, 17 colonnes sont ajoutées, nommez-les comme suit :

6. Pour chacune des colonnes sélectionnez le **Type** : chaîne de caractères (**String**).

7. Cliquez sur **OK**.

8. Dans l'espace de modélisation, double-cliquez sur le composant **tLogRow** pour afficher sa vue **Component**.

9. Cliquez sur le bouton **[...]** du champ **Edit schema** pour vérifier que le schéma a bien été propagé. Si ce n'est pas le cas, cliquez sur le bouton **Sync columns**.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** pour l'exécuter.

```
n|n|n|y|y|y|y|y|n|y|n|y|y|y|n|y|republican
n|n|n|n|n|n|y|y|y|y|n|n|y|n|y|y|democrat
y|n|y|n|n|n|y|y|y|y|n|y|n|n|y|y|democrat
n|n|y|y|y|y|n|n|y|y|n|y|y|y|n|y|republican
n|n|y|n|n|n|y|y|y|y|n|n|n|n|n|y|democrat
n|n|n|y|y|y|n|n|n|n|y|y|y|y|n|y|republican
n|n|n|y|y|y|n|n|n|n|n|y|y|y|n|y|republican
n|y|n|y|y|y|n|n|n|y|n|y|y|y|n|n|republican
[statistics] disconnected
Job Test ended at 13:36 16/11/2009. [exit code=0]
```

La console affiche les données contenues dans le fichier ARFF délimitées par une barre verticale (séparateur par défaut).

tFileInputDelimited

Propriétés du tFileInputDelimited

Famille de composant	File/Input	
Fonction	Le composant **tFileInputDelimited** lit un fichier ou un flux de données ligne par ligne avec des champs simples séparés. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job Map/Reduce, afin de générer du code Map/Reduce. Pour plus d'informations, consultez *tFileInputDelimited in Talend Map/Reduce Jobs*.	
Objectif	Ce composant lit un fichier ou un flux de données ligne par ligne, afin de le diviser en champs et d'envoyer ses champs au composant suivant, comme défini par le schéma, via une connexion Row.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File name/Stream	**File name** : Chemin d'accès et nom du fichier à traiter. **Stream** : Flux de données à traiter. Les données doivent préalablement être mises en flux afin d'être récupérées par le **tFileInputDelimited** via la variable représentative correspondante. Cette variable peut être prédéfinie dans votre Studio ou fournie par le contexte ou les composants utilisé(s) avec ce composant. Sinon, vous pouvez la définir manuellement et l'utiliser selon votre Job, par exemple à l'aide d'un **tJava** ou d'un **tJavaFlex**. Afin d'éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d'autocomplétion (**Ctrl+Espace**) afin de remplir le champ, si cette variable a été correctement définie. Pour plus d'informations concernant les variables disponibles, consultez le *Guide utilisateur du Studio Talend*.
	Row separator	Saisissez le séparateur à utiliser pour identifier la fin des lignes.
	Field separator	Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.
	CSV options	Cochez cette case pour prendre en compte les paramètres spécifiques aux fichiers CSV, notamment la manière de protéger les caractères dans le champ **Escape char** et le type de guillemet dans le champ **Enclosure**.
	Header	Saisissez le nombre de lignes à ignorer au début du fichier.
	Footer	Nombre de lignes à ignorer à la fin d'un fichier.
	Limit	Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n'est lue ni traitée.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository** Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Notez que, si la valeur d'entrée d'un champ primitif non nullable est nulle, la ligne de données comprenant ce champ sera rejetée.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Skip empty rows	Cochez cette case pour ignorer les lignes vides.
	Uncompress as zip file	Cochez cette case pour décompresser le fichier d'entrée.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**. Pour capturer l'exception `FileNotFoundException`, vous devez cocher cette case.
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Extract lines at random	Cochez cette case pour définir le nombre de lignes à extraire de façon aléatoire.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
	Trim all columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes.
	Check each row structure against schema	Cochez cette case pour vérifier si le nombre total de colonnes pour chaque ligne correspond au schéma. S'il ne correspond pas, un message d'erreur s'affichera dans la console.
	Check date	Cochez cette case afin de vérifier le format de la date strictement par rapport au schéma d'entrée
	Check columns to trim	Cochez la case devant le nom de chacune des colonnes dont vous voulez supprimer les espaces de début et de fin de champ.
	Split row before field	Cochez cette case pour séparer les lignes avant de séparer les champs.
	Permit hexadecimal (0xNNN) or octal (0NNNN) for numeric types	Cochez cette case si l'un de vos types numériques (long, integer, short, ou byte), doit être parsé depuis une chaîne de caractères hexadécimale ou octale. Cochez cette case à côté de la ou des colonne(s) d'intérêt pour transformer la chaîne de caractères d'entrée de chaque colonne sélectionnée en un type de données qui est celui défini dans le schéma. Cochez la case **Permit hexadecimal or octal** pour sélectionner toutes les colonnes. Cette table apparaît uniquement lorsque la case **Permit hexadecimal (0xNNN) or octal (0NNNN) for numeric types** est cochée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour lire un fichier et séparer les champs à l'aide du séparateur spécifié. Il permet de créer un flux de données à l'aide d'un lien **Row** > **Main**, ainsi que de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d'utilisation de ces deux liens, consultez *Scénario 2 : Extraire les données XML erronées dans un flux de rejet* du composant **tFileInputXML**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

tFileInputDelimited in Talend Map/Reduce Jobs

*Les informations contenues dans cette section concernent uniquement les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend** et ne sont pas applicables aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job Map/Reduce **Talend**, le **tFileInputDelimited**, ainsi que le Job complet utilisant ce composant, génère du code Map/Reduce natif. Cette section présente les propriétés spécifiques du **tFileInputDelimited** lorsqu'il est utilisé dans un Job Map/Reduce. Pour plus d'informations concernant un Job Map/Reduce **Talend**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	MapReduce / Input	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Réutilisez les propriétés stockées centralement sous le nœud **Hadoop Cluster** de la vue **Repository**. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Pour plus d'informations concernant le nœud **Hadoop Cluster**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si vous effectuez des modifications, le schéma devient automatiquement built-in.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Folder/File	Parcourez votre système ou saisissez le chemin d'accès au répertoire de HDFS dans lequel sont stockées les données à utiliser. Si le chemin d'accès défini pointe vers un dossier, ce composant lit tous les fichiers stockés dans le dossier, par exemple */user/talend/in*. Si des sous-dossiers existent, ils sont automatiquement ignorés à moins que le chemin défini soit */user/talend/in/**. Si vous souhaitez spécifier plusieurs fichiers ou dossiers dans ce champ, séparez les chemins à l'aide d'une virgule (,). Si le fichier à lire est un fichier compressé, saisissez son nom ainsi que son extension. Le **tHDFSInput** décompresse le fichier automatiquement à l'exécution. Les formats de compression, ainsi que les extensions correspondantes, sont : • DEFLATE : **.deflate* • gzip : **.gz* • bzip2 : **.bz2* • LZO : **.lzo* Assurez-vous que la connexion à la distribution Hadoop utilisée est correctement configurée dans l'onglet **Hadoop configuration** de la vue **Run**.
	Die on error	Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	Row separator	Saisissez le séparateur à utiliser pour identifier la fin des lignes.
	Field separator	Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.
	Header	Saisissez le nombre de lignes à ignorer au début du fichier.
	CSV options	Cochez cette case pour inclure des paramètres CSV spécifiques comme **Escape char** et **Text enclosure**.
	Skip empty rows	Cochez cette case pour ignorer les lignes vides.
Advanced settings	*Custom Encoding*	Il est possible de rencontrer des problèmes d'encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste **Encoding**. Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement.
	Advanced separator (for number)	Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.).
	Trim all columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées. Lorsque cette case est cochée, la table **Check column to trim** est affichée, ce qui vous permet de sélectionner les colonnes desquelles vous souhaitez enlever les espaces en début et en fin de champ.
	Check column to trim	Cette table est automatiquement renseignée avec le schéma utilisé. Cochez la (les) case(s) correspondant à la (aux) colonne(s) dont vous souhaitez supprimer les espaces et début et en fin de champ.
	Check each row structure against schema	Cochez cette case afin de vérifier que le nombre total de colonnes est cohérent par rapport au schéma. Si le nombre n'est pas cohérent, un message d'erreur s'affiche dans la console.
	Check date	Cochez cette case pour vérifier strictement le format de date par rapport au schéma d'entrée.

	Decode String for long, int, short, byte Types	Cochez cette case si l'un de vos types numériques (long, integer, short, ou byte), doit être parsé depuis une chaîne de caractères hexadécimale ou octale.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme composant de début et requiert un composant de transformation comme lien de sortie. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Une fois qu'un Job Map/Reduce est ouvert dans l'espace de modélisation graphique, le **tFileInputDelimited** et toute la famille MapReduce apparaissent dans la **Palette** du *Studio Talend*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.	

Scénario 1 : Afficher le contenu d'un fichier délimité

Le scénario suivant est un Job de deux composants ayant pour objectif de lire les lignes d'un fichier, de sélectionner des données délimitées et d'afficher la sortie dans la console de la vue **Run**.

Déposer et relier les composants

1. Cliquez et déposez un composant **tFileInputDelimited** et un composant **tLogRow** de la **Palette** dans l'espace de modélisation.

2. Cliquez-droit sur le composant **tFileInputDelimited** et sélectionnez **Row** > **Main** dans le menu contextuel. Puis glissez ce lien vers le **tLogRow** et relâchez lorsque le symbole de prise de courant apparaît.

Configurer les composants

1. Sélectionnez le composant **tFileInputDelimited** à nouveau et définissez ses propriétés dans l'onglet **Basic settings** :

2. Renseignez le chemin d'accès au fichier dans le champ **File Name**. Ce champ est obligatoire.

   Si le chemin d'accès du fichier contient des caractères accentués, vous obtiendrez un message d'erreur lors de l'exécution du Job. Pour plus d'informations concernant la procédure à suivre lorsque les accents ne sont pas supportés, consultez leGuide d'installation et de migration Talend.

3. Définissez le séparateur de lignes dans le champ **Row separator** afin d'identifier la fin d'une ligne. Puis définissez le séparateur de champs dans **Field separator** pour délimiter les champs composant une ligne.

4. Dans ce scénario, l'en-tête (**Header**) et le pied de page (**Footer**) n'ont pas besoin d'être définis. Et la limite de lignes lues (**Limit**) est de 50 pour cet exemple.

5. Sélectionnez soit local (**Built-in**) soit distant (**Repository**) comme **Schema** pour définir les données qui passent par le composant **tLogRow**.

6. Vous pouvez charger et/ou éditer le schéma à l'aide de la fonction **Edit Schema**.

   Voir également le *Guide utilisateur du Studio Talend*.

7. Saisissez l'encodage utilisé dans le fichier d'entrée. Ce paramètre permet d'assurer l'homogénéité de l'encodage dans tous les fichiers d'entrée et de sortie.

8. Sélectionnez le composant **tLogRow** et définissez le séparateur de champs de la sortie affichée. Voir également *Propriétés du tLogRow*.

9. Cochez la case **Print schema column name in front of each value** pour récupérer le libellé des colonnes dans la sortie affichée.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Cliquez sur la vue **Run**, puis cliquez sur **Run** pour exécuter le Job.

   Le fichier est lu ligne par ligne et les champs extraits sont affichés dans la console, tel que défini dans les propriétés du composant (dans l'onglet **Basic settings**).

```
Starting job FileDel at 10:45 12/04/2007.
ID_Owner: 2|Name: lebouh|ID_Insurance: PKI2906
ID_Owner: 3|Name: bouhnan|ID_Insurance: BNU9147
ID_Owner: 4|Name: hirtle|ID_Insurance: TEV8360
ID_Owner: 5|Name: bobouh|ID_Insurance: WPM3151
ID_Owner: 6|Name: carbone|ID_Insurance: IFY9885
```

La console affiche l'en-tête suivi des données lues par le Job.

Scénario 2 : Lire les données d'un fichier distant en mode stream

Le scénario suivant illustre un Job à quatre composants ayant pour objectif de récupérer les données d'un fichier volumineux quasiment aussitôt qu'elles ne sont lues afin d'éviter l'attente du téléchargement de l'ensemble du fichier de données, et les affichent dans la console de la vue **Run**.

Déposer et relier les composants

1. Déposez les composants suivants dans l'espace de modélisation : **tFileFetch**, **tSleep**, **tFileInputDelimited**, et **tLogRow**.
2. Reliez les composants **tSleep** et **tFileInputDelimited** via un lien de type **Trigger** > **OnComponentOk** et reliez les composants **tFileInputDelimited** et **tLogRow** via un lien de type **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le composant **tFileFetch** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.

2. Dans la liste **Protocol**, sélectionnez le type de protocole vous permettant d'accéder au serveur sur lequel est stocké votre fichier de données.

3. Dans le champ **URI**, saisissez l'URI d'accès au serveur sur lequel est stocké votre fichier.

4. Cochez la case **Use cache to save the resource** pour mettre les données de votre fichier en mémoire cache. Cette option permet d'utiliser le mode stream pour transférer les données du fichier.

5. A partir de l'espace de modélisation, double-cliquez sur le composant **tSleep** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.

   Par défaut, le champ **Pause** du composant **tSleep** correspond à *1* seconde. Laissez ce paramètre par défaut. Il met le deuxième Job en pause afin de laisser le temps au premier Job contenant le **tFileFetch** de lire les données du fichier.

6. A partir de l'espace de modélisation, double-cliquez sur le composant **tFileInputDelimited** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.

7. Dans le champ **File name/Stream** :

   - Effacez le contenu par défaut.

   - Faites un **Ctrl+Espace** pour faire apparaître la liste des variables disponibles à partir de ce composant.

   - Sélectionnez la variable *tFileFetch_1_INPUT_STREAM* dans la liste d'auto-complétion, la variable suivante apparaît dans le champ **Filename** : `((java.io.InputStream)globalMap.get("tFileFetch_1_INPUT_STREAM"))`.

8. Dans la liste **Schema**, sélectionnez **Built-in** et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour décrire la structure des données du fichier que vous souhaitez récupérer. Le fichier *US_Employees* est composé de six colonnes : *ID*, *Employee*, *Age*, *Address*, *State*, *EntryDate*.

   Cliquez sur le bouton **[+]** pour ajouter les six colonnes et paramétrez-les comme indiqué sur la capture d'écran ci-dessus. Cliquez sur **OK**.

9. A partir de l'espace de modélisation, double-cliquez sur le composant **tLogRow** pour afficher l'onglet **Basic settings** de la vue **Component** et cliquez sur **Sync Columns** pour vous assurer que le schéma est récupéré du composant précédent.

Configurer l'exécution du Job et exécuter le Job

1. A partir de l'espace de modélisation, cliquez sur la vue **Job** et sur l'onglet **Extra**.

2. Cochez la case **Multi thread execution** afin d'exécuter les deux Jobs en parallèle, sachant que le deuxième Job a un retard d'une seconde par rapport au premier comme défini dans le **tSleep**. Cette option permet donc de récupérer les données grâce au **tFileDelimited** quasiment dès leur lecture par le **tFileFetch**.

3. Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
105730|Harry QUINCY|32|Woodson Rd., Pierre|FLORIDA|2008-03-09
105731|Ronald EISENHOWER|26|East Calle Primera, Saint
Paul|TEXAS|2008-07-03
105732|Ulysses BUCHANAN|32|North Ventu Park Road,
Lansing|ARKANSAS|2008-02-16
105733|Thomas TAFT|45|Moreno Drive, Boise|WEST VIRGINIA|2008-08-08
105734|John EISENHOWER|45|Steele Lane, Indianapolis|ALASKA|2007-07-17
105735|Andrew QUINCY|33|El Camino Real, Boston|WISCONSIN|2008-08-25
105736|Lyndon ADAMS|49|Harbor Dr, Frankfort|MARYLAND|2008-12-28
105737|George EISENHOWER|38|Monroe Street, Sacramento|UTAH|2008-09-04
105738|Grover EISENHOWER|42|Steele Lane, Pierre|NEVADA|2008-01-17
105739|Chester CARTER|39|Fontaine Road, Augusta|WEST VIRGINIA|2007-12-20
105740|Abraham NIXON|30|Monroe Street, Columbia|NEBRASKA|2008-07-29
Job tFileInputDelimited ended at 17:58 09/02/2010. [exit code=0]
```

Les données sont affichées dans la console dès leur lecture.

tFileInputEBCDIC

Ce composant requiert l'utilisation d'une JDK Oracle.

Propriétés du tFileInputEBCDIC

Famille de composant	File/Input	Notez que ce composant est déprécié. Les capacités de traitement EBCDIC sont désormais entièrement supportées par *Talend Data Mapper*, qui est inclus avec n'importe quelle solution de **Talend** Platform. Pour plus d'informations concernant *Talend Data Mapper*, consultez le *Guide utilisateur de Talend Data Mapper*.
Fonction	Le composant **tFileInputEBCDIC** lit un fichier EBCDIC et extrait les données en fonction du schéma sélectionné.	
Objectif	Ce composant ouvre un fichier et le lit afin de séparer les données en fonction des schémas décrivant la structure de ce fichier et d'envoyer ses données et métadonnées au(x) composant(s) suivant(s) via une connexion **Row** > **Main**.	
Basic settings	*Property type*	Peut être Built-in ou Repository.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema(s)	Cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) et cliquez sur le bouton **[...]** dans la colonne **Schema** d'une ligne sélectionnée afin de définir le schéma des données à traiter. Le nom du schéma est modifiable et s'affiche dans la colonne **Distinguish field value** de la ligne sélectionnée.
	Data file	Cliquez sur le bouton **[...]** afin de parcourir votre système, ou saisissez le chemin d'accès au fichier EBCDIC contenant les données à traiter.
	Edit schema	Cliquez sur le bouton **[...]** afin d'éditer le schéma des données à traiter.
		Built-in : Sélectionnez cette option pour éditer le schéma Built-in des données à traiter.
		Repository : Sélectionnez cette option pour éditer le schéma du **Repository** sélectionné. Les champs suivants sont automatiquement renseignés à l'aide du schéma sélectionné.
	Xc2j file	Cliquez sur le bouton **[...]** afin de parcourir votre système, ou saisissez le chemin d'accès au fichier xc2j pour transformer le(s) schéma(s) EBCDIC en un fichier XML intermédiaire. Ce champ est désactivé et le fichier xc2j n'est pas requis lorsque vous cochez la case **Custom set Original Length in Schema**.
	Custom set Original Length in Schema	Cochez cette case pour augmenter la vitesse de lecture des fichiers. Lorsque vous cochez cette case, le champ **Xc2j file** est désactivé et le fichier xc2j n'est pas nécessaire. Vous pouvez éditer le schéma Built-in ou Repository pour les données à traiter.
Advanced settings	*Encoding*	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Trim all column	Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes définies.

	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour lire un fichier EBCDIC et séparer les données en fonction des schémas identifiés dans ce fichier.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Extraire les données d'un fichier EBCDIC

Ce scénario décrit un Job qui lit un fichier EBCDIC, en extrait les données puis les affiche selon ses trois différents schémas.

Ce scénario utilise l'assistant **[EBCDIC Connection]** qui guide l'utilisateur à travers plusieurs étapes afin de créer une connexion à un fichier Copybook qui décrit la structure du fichier EBCDIC correspondant et récupère ses différents schémas. Cet assistant est uniquement disponible pour les utilisateurs ayant souscrit à l'une des solutions *Talend*.

Dans ce scénario, la connexion EBCDIC est appelée *ebcdic* et contient trois différents schémas appelés *_0*, *_1* et *_2* qui sont utilisés dans les composants tExtractEBCDICFields dans ce Job.

Déposer les composants

- Créez un Job et ajoutez les composants suivants en saisissant leurs noms dans l'espace de modélisation graphique ou en les déposant de la **Palette** : un composant **tFileInputEBCDIC**, un composant **tMap**, trois composants **tExtractEBCDICFields** et trois composants **tLogRow**.

Configurer le composant d'entrée

1. Double-cliquez sur le composant **tFileInputEBCDIC** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Custom set Original Length in Schema**.

3. Cliquez sur le bouton **[...]** à côté de **Edit schema** pour ouvrir l'éditeur de schéma du **tFileInputEBCDIC**.

4. Cliquez sur le bouton **[+]** afin d'ajouter deux colonnes *recid* et *filler* pour extraire le champ ID de chaque schéma dans la colonne *recid* et de laisser les autres champs dans la colonne *filler*.

   Notez que vous devez paramétrer pour chaque colonne les bonnes valeurs pour les champs suivants :

   - **Type** : Sélectionnez le type de colonne de schéma.
   - **DB Type** : Sélectionnez le type du champ correspondant dans le fichier EBCDIC.
   - **Length** : Spécifiez la longueur de la colonne de schéma.
   - **OriginalLength** : Spécifiez la longueur du champ correspondant dans le fichier EBCDIC. Dans cet exemple, *1099* est la longueur totale des autres champs à l'exclusion du champ ID dans chaque schéma.
   - **ImpliedDecimal** : Renseignez **true** ou **false** pour indiquer s'il y a, dans le fichier EBCDIC, une virgule implicite à un emplacement défini dans un champ.
   - **Signed** : Renseignez **true** ou **false** pour indiquer si un champ est signé ou non dans le fichier EBCDIC.
   - **Precision** : Spécifiez le nombre de chiffres à droite de la virgule.

5. Cliquez sur **OK** pour enregistrer les paramètres et fermer l'éditeur de schéma.

6. Cliquez sur le bouton **[...]** à côté du champ **Data file** et parcourez votre système jusqu'au fichier EBCDIC contenant les données à traiter : *E:/tFileInputEBCDIC/ebcdic_data.txt* dans cet exemple.

7. Connectez le **tFileInputEBCDIC** au **tMap** à l'aide d'une connexion de type **Row** > **Main**.

Configurer le composant tMap

1. Cliquez-droit sur le **tMap** et sélectionnez **Row** > ***New Ouput* (Main)** dans le menu contextuel, puis cliquez sur le premier **tExtractEBCDICFields**. Saisissez *rec1* dans la boîte de dialogue qui s'ouvre et cliquez sur **OK** pour connecter le **tMap** au premier **tExtractEBCDICFields** à l'aide d'un lien appelé *rec1*.

   Répétez cette étape pour connecter le **tMap** aux deuxième et troisième composants **tExtractEBCDICFields** à l'aide de liens respectifs *rec5* et *rec9*.

2. Double-cliquez sur le **tMap** pour ouvrir l'éditeur de map.

3. Cliquez-déposez toutes les colonnes de la table d'entrée *row1* vers leurs tables de sortie respectives *rec1*, *rec5* et *rec9*.

4. Cliquez sur le bouton dans le coin supérieur droit de la table de sortie *rec1* pour activer le filtre d'expression, puis renseignez-le avec la condition *row1.recid.equals("1")*.

   Répétez cette étape pour activer le filtre d'expression des tables de sortie *rec5* et *rec9*, puis renseignez-les de manière respective avec les conditions *row1.recid.equals("5")* et *row1.recid.equals("9")*. Les données avec trois schémas différents seront ainsi séparées en trois destinations respectives selon leur valeur ID.

5. Cliquez sur **OK** pour enregistrer les paramètres et fermer l'éditeur de map.

Configurer les composants tExtractEBCDICFields

1. Double-cliquez sur le premier **tExtractEBCDICFields** pour ouvrir sa vue **Basic settings**.

2. Sélectionnez **Repository** dans la liste **Schema** puis cliquez sur le bouton **[...]** pour ouvrir la boîte de dialogue **[Repository Content]** dans laquelle vous pouvez sélectionnez le schéma du référentiel pour ce composant.

3. Sélectionnez le schéma *_0* sous la connexion *ebcdic* pour le premier **tExtractEBCDICFields** puis cliquez sur **OK** pour enregistrer les paramètres et fermer la boîte de dialogue.

4. Cliquez sur le bouton **[...]** à côté de **Edit schema**, sélectionnez **View schema** dans la boîte de dialogue qui s'ouvre puis cliquez sur **OK** pour ouvrir l'éditeur de schéma.

5. Décochez la case dans **Used Column** afin de supprimer du schéma la colonne *HDRRECID* qui a déjà été extraite du **tFileInputEBCDIC**, puis cliquez sur **OK** pour enregistrer les paramètres et fermer l'éditeur de schéma.

6. Sélectionnez *filler* dans la liste **Field** pour extraire les données de ce champ en encodage EBCDIC.

7. Répétez les étapes précédentes, de **Step 1** à **Step 6**, afin de paramétrer le schéma du référentiel *_1* pour le deuxième **tExtractEBCDICFields**,supprimer la colonne *DTLRECID* du schéma et sélectionner *filler* dans la liste **Field**.

8. Répétez les étapes précédentes, de **Step 1** à **Step 6**, afin de paramétrer le schéma du référentiel *_2* pour le troisième **tExtractEBCDICFields**, supprimer la colonne *TRLRECORDID* du schéma et sélectionner *filler* dans la liste **Field**.

9. Connectez chaque **tExtractEBCDICFields** à chaque **tLogRow** à l'aide de connexions respectives de type **Row** > **Main**.

Configurer les composants tLogRow

1. Double-cliquez sur le premier **tLogRow** afin d'ouvrir sa vue **Basic settings**.

2. Cochez la case **Print component unique name in front of each output row** pour un meilleur affichage des résultats.

3. Répétez les deux étapes précédentes pour configurer les deuxièmes et troisième composants **tLogRow**.

Enregistrer et exécuter le Job

1. Appuyez sur **Ctrl+S** pour enregistrer votre Job.

2. Appuyez sur **F6** pour exécuter le Job.

```
Starting job tFileInputEBCDIC_Demo at 11:37 19/08/2014.

[statistics] connecting to socket on port 3675
[statistics] connected
[tLogRow_1] 9330|B9330ATD|20140310|1|1|
[tLogRow_2] 9330|6|0|7985989|2014041|13|0| |
|2014011|0|2014041|0|1|2014021|1200012|B22QSZF6TKRQPVVW|0| |C|
|4264290011289089   |4264290011289089   |24692164020000056439303| |DBT| | | |
| | | | |VI|Amazon.com                AMZN.COM/BILLWA |US |                |N|
|Z|7| | |015976|5|9|01|4|784959000762203|A|0|D| | | | | | | | | | |W|PR|
|   |584020515|7784|39|840|00|    |Y| | |G| |N| | | |
|0|0|0|0|98101    |5942|1001|1|1|10025426|0|0|840|3|840|2|2|11|              |
| |    | | |V| |0|
|2014021|92.07|0|0|2014020|9207|92.07|0.00|2014021|4021|9207|0|0|0.000|0.0000|
|0|N| | | |    | |0.00| |     |     |0|0.00|   |00006|   |    | |
|   | |0|   | |  |USD| | | |0| |0|1.000000000|0.000000000|0|0.00|    |    |  |
| | | | |        |    | |          | |0.00|0|0.00|0.00|0.00| |0.00|0|0|0|0| |
|
[tLogRow_3] 9330|B9330ATD|5|6654000.08|0.00|468200000.00|0.00|
[statistics] disconnected
Job tFileInputEBCDIC_Demo ended at 11:37 19/08/2014. [exit code=0]
```

Les données extraites sont affichées dans la console.

tFileInputExcel

Propriétés du tFileInputExcel

Famille du composant	File/Input	
Fonction	Le composant **tFileInputExcel** lit un fichier Excel (.xls ou .xlsx) ou un flux de données et en extrait les données ligne par ligne.	
Objectif	Ce composant lit un fichier Excel ou un flux de données ligne par ligne pour le scinder en champs et envoie les champs tel que définis dans le schéma au composant suivant du Job via une connexion **Row**.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de connexion et enregistrer les paramètres de connexion au fichier Excel que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de fichier, consultez le *Guide utilisateur du Studio Talend*.
	Read excel2007 file format (xlsx)	Cochez cette case pour lire un fichier .xlsx d'Excel 2007.
	File name/Stream	**File name** : Chemin d'accès et nom du fichier, et/ou variable à traiter. **Stream** : Flux de données à traiter. Les données doivent préalablement être mises en flux afin d'être récupérées par le **tFileInputExcel** via la variable *INPUT_STREAM* disponible dans la liste d'auto-complétion (**Ctrl+Espace**). Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	All sheets	Cochez cette case pour traiter toutes les feuilles du fichier Excel.
	Sheet list	Cliquez sur le bouton **[+]** pour ajouter de nouvelles lignes à la liste de feuilles à traiter : **Sheet (name or position)** : renseignez le nom ou la position de la feuille à traiter. **Use Regex** : cochez cette case si vous utilisez une expression régulière pour filtrer les feuilles selon un modèle.
	Header	Nombre de lignes à ignorer au début d'un fichier.
	Footer	Nombre de lignes à ignorer à la fin d'un fichier.
	Limit	Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n'est lue ni traitée.
	Affect each sheet(header&footer)	Cochez cette case pour appliquer les paramètres définis dans les champs **Header** et **Footer** à toutes les feuilles traitées.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les

		lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	First column et *Last column*	Renseignez l'intervalle de colonnes que vous voulez traiter : la position de la première colonne dans le champ **First column** et celle de la dernière colonne à traiter dans le champ **Last column**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Trim all columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées. Lorsque cette case est cochée, la table **Check column to trim** est affichée, ce qui vous permet de sélectionner les colonnes desquelles vous souhaitez enlever les espaces en début et en fin de champ.
	Check column to trim	Cette table est automatiquement renseignée avec le schéma utilisé. Cochez la (les) case(s) correspondant à la (aux) colonne(s) dont vous souhaitez supprimer les espaces et début et en fin de champ.
	Convert date column to string	Disponible lorsque la case **Read excel2007 file format (xlsx)** est cochée, dans la vue **Basic settings**. Cochez cette case pour afficher la table **Check need convert date column**. Vous pouvez convertir les colonnes Excel contenant des dates en colonnes de type String dans le fichier de sortie, avec le modèle défini. **Column** : toutes les colonnes disponibles dans le schéma du fichier source .xlsx. **Convert** : cochez cette case pour choisir toutes les colonnes à convertir (si elles sont de type String). Vous pouvez également cocher la case de chaque colonne à convertir. **Date pattern** : définissez le format de date.

	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	Read real values for numbers	Cochez cette case pour lire les nombres en valeur réelle. Cette case est indisponible lorsque vous cochez la case **Read excel2007 file format (xlsx)** dans la vue **Basic settings**.
	Stop reading on encountering empty rows	Cochez cette case pour ignorer les lignes vides, et, s'il y en a, les lignes qui suivent. Cette case est indisponible lorsque vous cochez la case **Read excel2007 file format (xlsx)** dans la vue **Basic settings**.
	Generation mode	Disponible lorsque la case **Read excel2007 file format (xlsx)** est cochée dans la vue **Basic settings**. Sélectionnez le mode de lecture du fichier Excel 2007. • **Less memory consumed for large excel(Event mode)** : utilisé pour un fichier volumineux. Ce mode économise de la mémoire car il permet de lire le fichier Excel 2007 comme un flux. • **Memory-consuming (User mode)** : utilisé pour un petit fichier. Nécessite beaucoup de mémoire.
	Don't validate the cells	Cochez cette case pour ne pas effectuer de validation des données. Cette case est indisponible lorsque vous cochez la case **Read excel2007 file format (xlsx)** dans la vue **Basic settings**.
	Ignore the warning	Cochez cette case pour ignorer les avertissements générés par des erreurs sur le fichier Excel. Cette case est indisponible lorsque vous cochez la case **Read excel2007 file format (xlsx)** dans la vue **Basic settings**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **CURRENT_SHEET** : nom de la feuille traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour lire un fichier Excel (.xls ou .xlsx) et séparer les champs en fonction d'un schéma défini. Ce composant permet de créer un flux de données à l'aide d'un lien **Row** > **Main**, ainsi que de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d'utilisation de ces deux liens, consultez *Scénario 2 : Extraire les données XML erronées dans un flux de rejet* du composant **tFileInputXML**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tFileInputFullRow

Propriétés du tFileInputFullRow

Famille de composant	File/Input	
Fonction	Le composant **tFileInputFullRow** lit un fichier donné ligne par ligne. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job Map/Reduce **Talend** afin de générer du code Map/Reduce. Pour plus d'informations, consultez *Le tFileInputFullRow dans des Jobs Map/Reduce Talend*.	
Objectif	Le **tFileInputFullRow** ouvre un fichier qu'il lit ligne par ligne et passe les lignes complètes au composant suivant comme défini dans le schéma, via une connexion de type Row.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	File Name	Chemin d'accès et nom du fichier, et/ou variable à traiter. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Row separator	Saisissez le séparateur à utiliser pour identifier la fin des lignes.
	Header	Saisissez le nombre de lignes à ignorer au début du fichier.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Header	Nombre de lignes à ignorer en début de fichier.
	Footer	Nombre de lignes à ignorer à la fin d'un fichier.
	Limit	Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n'est lue ou traitée.
	Skip empty rows	Cochez cette case pour ignorer les lignes vides.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
Advanced settings	*Encoding*	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.

	Extract lines at random	Cochez cette case pour définir le nombre de lignes à extraire de façon aléatoire.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour lire des lignes complètes dans un fichier délimité pouvant être très gros. Ce composant permet de créer un flux de données à l'aide d'un lien **Row** > **Main**, ainsi que de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d'utilisation de ces deux liens, consultez *Scénario 2 : Extraire les données XML erronées dans un flux de rejet* du composant **tFileInputXML**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Le tFileInputFullRow dans des Jobs Map/Reduce Talend

*Les informations contenues dans cette section concernent uniquement les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend** et ne sont pas applicables aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job Map/Reduce **Talend**, le **tFileInputFullRow** et le Job Map/Reduce complet l'utilisant génèrent nativement du code Map/Reduce. Cette section présente les propriétés spécifiques du **tFileInputFullRow** dans cette situation. Pour plus d'informations concernant les Jobs Map/Reduce **Talend**, consultez *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	MapReduce / Input	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Réutilisez les propriétés stockées centralement sous le nœud **Hadoop Cluster** de la vue **Repository**. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Pour plus d'informations concernant le nœud **Hadoop Cluster**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si vous effectuez des modifications, le schéma devient automatiquement built-in.

		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Folder/File	Parcourez votre système ou saisissez le chemin d'accès au répertoire de HDFS dans lequel sont stockées les données à utiliser. Si le chemin d'accès configuré pointe vers un dossier, le composant lit tous les fichiers stockés dans ce dossier, par exemple */user/talend/in*. Si le sous-dossier existe, les sous-dossiers sont automatiquement ignorés, sauf si vous définissez le chemin d'accès comme suit : */user/talend/in/**. Si vous souhaitez spécifier plus d'un fichier ou répertoire dans ce champ, séparez chaque chemin d'accès par une virgule (,). Si le fichier à lire est un fichier compressé, saisissez son nom et son extension. Le **tHDFSFullRow** décompresse automatiquement le fichier lors de l'exécution. Les formats de compression supportés et leur extension sont : • DEFLATE : **.deflate* • gzip : **.gz* • bzip2 : **.bz2* • LZO : **.lzo* Assurez-vous que la connexion à la distribution Hadoop utilisée est correctement configurée dans l'onglet **Hadoop configuration** de la vue **Run**.
	Die on error	Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	Row separator	Saisissez le séparateur à utiliser pour identifier la fin des lignes.
	Header	Saisissez le nombre de lignes à ignorer au début du fichier.
	Skip empty rows	Cochez cette case pour ignorer les lignes vides.
Advanced settings	*Custom Encoding*	Il est possible de rencontrer des problèmes d'encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste **Encoding**. Sélectionnez l'encodage à utiliser dans la liste ou sélectionnez **Custom** pour le définir manuellement.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Usage	Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme composant de début et requiert un composant de transformation comme lien de sortie. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop.	

	Lorsqu'un Job Map/Reduce est ouvert dans l'espace de modélisation graphique, le **tFileInputFullRow** et la famille **MapReduce** s'affichent dans la **Palette** du Studio. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.

Scénario : Lire des lignes complètes dans un fichier délimité

Le scénario suivant crée un Job à deux composants permettant de lire des lignes complètes à partir d'un fichier et d'afficher les données de sortie dans la console de log **Run**.

1. A partir de la **Palette**, cliquez-déposez les composants **tFileInputFullRow** et **tLogRow** dans l'éditeur graphique.

2. Cliquez-droit sur le composant **tFileInputFullRow** et connectez-le au **tLogRow** à l'aide d'un lien de type **Row Main**.

3. Dans l'éditeur graphique, sélectionnez le composant **tFileInputFullRow**.

4. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tFileInputFullRow**.

5. Dans l'onglet **Basic settings**, configurez le champ **Schema** en mode **Built-In**.

6. Cliquez sur le bouton **[...]** du champ **Edit schema** pour voir les données à passer au composant **tLogRow**. Notez que le schéma est en lecture seule et ne comporte qu'une seule colonne, *line*.

7. Dans le champ **File Name**, renseignez le chemin d'accès au fichier à traiter ou cliquez sur le bouton **[...]**. Ce champ est obligatoire. Dans ce scénario, le fichier à lire s'appelle *test5* et comporte trois lignes comprenant chacune deux champs séparés par un point-virgule.

8. Dans le champ **Row separator**, renseignez le séparateur permettant d'identifier les fins de ligne.

9. Dans la champ **Header**, indiquez la valeur 1. Dans ce scénario, ni le pied-de-page ni le nombre de lignes à traiter ne sont renseignés.

10. Dans l'éditeur graphique, sélectionnez le **tLogRow** puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**). Pour plus d'informations, consultez *tLogRow*.

11. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

12. Cliquez sur la vue **Run**, puis cliquez sur **Run** pour exécuter le Job.

```
Starting job Input_Full_Row at 11:13 26/08/2008.
.--------------------.
|      tLogRow_1     |
|=-------------------=|
|line                |
|=-------------------=|
|Janet,Anderson;I9988|
|Martin,Chairman;9889|
|Lily,Massy;9988     |
'--------------------'

Job Input_Full_Row ended at 11:14 26/08/2008. [exit
```

Le composant **tFileInputFullRow** lit les trois lignes une à une sans tenir compte des séparateurs de champs et les lignes complètes apparaissent dans la console **Run**.

Pour extraire des lignes uniquement les champs, utilisez le **tExtractDelimitedFields**, le **tExtractPositionalFields** et le **tExtractRegexFields**. Pour plus d'informations, consultez *tExtractPositionalFields*, *tExtractRegexFields*, .

tFileInputJSON

Propriétés du tFileInputJSON

Famille de composant	File/Input	
Fonction	Le composant **tFileInputJSON** extrait des données JSON selon la requête JSONPath. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job **Talend** Map/Reduce pour générer du code Map/Reduce. Pour plus d'informations, consultez *Le tFileInputJSON dans des Jobs Talend Map/Reduce*. Dans ce cas, le **tFileInputJSON** appartient à la famille de composants **MapReduce**.	
Objectif	Ce composant extrait des données JSON selon la requête JSONPath, transfère les données à un fichier, une table de base de données, etc.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Read by XPath	Cette case est cochée par défaut. Elle permet d'afficher le champ **Loop JSONPath query** et la case **Get nodes** dans la table **Mapping**.
	Use URL	Cochez cette case pour récupérer les données directement à partir d'internet.

		URL : entrez le chemin d'accès à l'URL à partir de laquelle récupérer les données.
	Filename	Ce champ est indisponible lorsque la case **Use URL** est cochée. Cliquez sur le bouton **[...]** situé à côté de ce champ afin de parcourir votre système jusqu'au fichier à partir duquel récupérer les données. Vous pouvez également directement saisir le chemin complet vers ce fichier.
	Loop JSONPath query	Requête JSONPath spécifiant le nœud de boucle des données JSON. Disponible lorsque la case **Read by XPath** est cochée.
	Mapping	**Column** : montre le schéma défini dans l'éditeur **Edit Schema**. **JSONPath Query** : Spécifiez le nœud JSON contenant les données souhaitées. Pour plus d'informations concernant les expressions JSONPath, consultez http://goessner.net/articles/JsonPath/ (en anglais). **Get nodes** : cochez cette case pour extraire les données JSON de tous les nœuds spécifiés dans la liste **Xpath query** ou cochez la case à côté d'un nœud spécifique pour extraire ses données JSON uniquement.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject** .
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Validate date	Cochez cette case pour vérifier le format de date strictement par rapport au schéma d'entrée. Cette case n'est disponible que si la case **Read By XPath** est cochée.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de départ. Il nécessite donc un lien de sortie.	
Utilisation dans des Jobs Map/Reduce	Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme composant de début et requiert un composant de transformation comme lien de sortie. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop.	

	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Le tFileInputJSON dans des Jobs Talend Map/Reduce

*Les informations contenues dans cette section concernant uniquement les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend** et ne s'appliquent pas aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job **Talend** Map/Reduce, le**tFileInputJSON**, ainsi que tout le Job Map/Reduce l'utilisant, génère nativement du code Map/Reduce. Cette section présente les propriétés spécifiques du **tFileInputJSON** lorsqu'il est utilisé dans cette situation. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	MapReduce / Input	
Fonction	Dans un Job Map/Reduce, le **tFileInputJSON** extrait des données d'un ou plusieurs fichier(s) JSON dans HDFS et les envoie dans le composant de transformation.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Permet de réutiliser les propriétés stockées centralement sous le nœud **File Json** du **Repository**. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Pour plus d'informations concernant le nœud **File Json**, consultez la section concernant la configuration d'un schéma de fichier JSON dans le *Guide utilisateur du Studio Talend*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Folder/File	Saisissez le chemin d'accès au fichier ou au dossier de HDFS duquel extraire les données. Si le chemin saisi pointe vers un dossier, tous les fichiers de ce dossier seront lus. Si le fichier à lire est un fichier compressé, saisissez son nom ainsi que son extension. Le **tFileInputJSON** décompresse le fichier automatiquement à l'exécution. Les formats de compression, ainsi que les extensions correspondantes, sont : • DEFLATE : **.deflate* • gzip : **.gz* • bzip2 : **.bz2* • LZO : **.lzo* Assurez-vous que la connexion à la distribution Hadoop utilisée est correctement configurée dans l'onglet **Hadoop configuration** de la vue **Run**.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	Loop Xpath query	Nœud du champ JSON sur lequel se base la boucle.
	Mapping	Renseignez la table **Mapping** pour extraire les données désirées. • **Column** : colonnes définies dans le schéma pour contenir les données extraites dans le champ JSON. • **XPath query** : requête XPath permettant de spécifier le nœud du champ JSON à extraire. • **Get Nodes** : cette case peut être cochée pour obtenir les valeurs d'un nœud dans un champ JSON.
Advanced settings	*Advanced separator (for number)*	Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.).
	Validate date	Cochez cette case pour vérifier le format de date strictement par rapport au schéma d'entrée.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Usage	Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme composant de début et requiert un composant de transformation comme lien de sortie. Les autres composants utilisés	

	avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Une fois qu'un Job Map/Reduce est ouvert dans l'espace de modélisation graphique, le **tFileInputJSON** et sa famille **MapReduce** apparaissent dans la **Palette** du studio. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.
Prérequis	La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d'exemple relatives à MapR. • Assurez-vous d'avoir installé le client MapR sur la même machine que le *Studio Talend* et d'avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D'après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées *MAPR_INSTALL\ hadoop\hadoop-VERSION\lib\native*. Par exemple, pour Windows, la bibliothèque est *lib\MapRClient.dll* dans le fichier Jar du client MapR. Pour plus d'informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n'ajoutez pas de librairie, il est possible que vous rencontriez l'erreur suivante : `no MapRClient in java.library.path`. • Configurez l'argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]**. Cet argument fournit au studio le chemin d'accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d'utiliser entièrement l'aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR. Pour plus d'informations concernant la configuration de cet argument, consultez la section expliquant comment visualiser les données dans le *Guide de prise en main de Talend Open Studio for Big Data*. Pour plus d'informations concernant l'installation d'une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénario 1 : Extraire des données JSON d'un fichier

Dans ce scénario, le composant **tFileInputJSON** lit des données JSON d'un fichier à l'aide de requêtes JSONPath et le composant **tLogRow** affiche les données plates extraites.

Le fichier JSON contient des informations concernant une vidéothèque.

Relier les composants

1. Déposez un **tFileInputJSON** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Renommez le **tFileInputJSON** en **read_JSON_data** et le **tLogRow** en **show_data**.
3. Reliez les composants à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFileInputJSON** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

3. Cliquez cinq fois sur le bouton **[+]** pour ajouter cinq colonnes, respectivement *type*, *movie_name*, *release*, *rating* et *starring*, toutes de type **String**, sauf la colonne *rating*, de type **Double**.

   Cliquez sur **OK** pour fermer l'éditeur.

4. Dans la boîte de dialogue **[Propagate]**, cliquez sur **Yes** pour propager le schéma aux composants suivants.

5. Dans le champ **Filename**, saisissez le chemin d'accès au fichier JSON.

   Dans cet exemple, le fichier JSON se présente comme suit :

```
{"movieCollection": [
```

```
    {
        "type": "Action Movie",
        "name": "Brave Heart",
        "details": {
            "release": "1995",
            "rating": "5",
            "starring": "Mel Gibson"
        }
    },
    {
        "type": "Action Movie",
        "name": "Edge of Darkness",
        "details": {
            "release": "2010",
            "rating": "5",
            "starring": "Mel Gibson"
        }
    }
]}
```

6. Décochez la case **Read By XPath**.

7. Dans la table **Mapping**, le schéma apparaît automatiquement dans la colonne **Column**.

Mapping

Column	JSONPath query
type	"$.movieCollection[*].type"
name	"$.movieCollection[*].name"
release	"$..release"
rating	"$..rating"
starring	"$..starring"

Dans la colonne **JSONPath query**, saisissez les requêtes suivantes :

- Dans les colonnes *type* et *name*, saisissez les requêtes JSONPath *"$.movieCollection[*].type"* et *"$.movieCollection[*].name"*, respectivement. Elles correspondent aux premiers nœuds des données JSON.

  Ici, *"$.movieCollection[*]"* fait référence au nœud racine relatif aux nœuds *type* et *name*, le nœud *movieCollection*.

- Pour les colonnes *release*, *rating* et *starring*, saisissez les requêtes JSONPath *"$..release"*, *"$..rating"* et *"$..starring"*, respectivement.

  Ici, *".."* fait référence à la descente récursive du nœud *details*, les nœuds *release*, *rating* et *starring*.

8. Double-cliquez sur le **tLogRow** pour afficher la vue **Basic settings** et sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.

```
[statistics] connecting to socket on port 3788
[statistics] connected
.------------+----------------+-------+------+----------.
|                        show_data                        |
|=-----------+----------------+-------+------+---------=|
|type        |name            |release|rating|starring  |
|=-----------+----------------+-------+------+---------=|
|Action Movie|Brave Heart     |1995   |5.0   |Mel Gibson|
|Action Movie|Edge of Darkness|2010   |5.0   |Mel Gibson|
'------------+----------------+-------+------+----------'
[statistics] disconnected
```

Comme affiché ci-dessus, les données source JSON sont collectées dans une table de fichier plat.

Scénario 2 : Extraire des données JSON d'un fichier en utilisant XPath

Dans ce scénario, le composant **tFileInputJSON** lit des données JSON d'un fichier à l'aide de requêtes XPath et le composant **tLogRow** affiche les données plates extraites.

Le fichier JSON contient des informations concernant une vidéothèque.

Relier les composants

1. Déposez un **tFileInputJSON** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Renommez le **tFileInputJSON** en **read_JSON_data** et le **tLogRow** en **show_data**.
3. Reliez les composants à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFileInputJSON** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

3. Cliquez cinq fois sur le bouton **[+]** pour ajouter cinq colonnes, respectivement *type*, *movie_name*, *release*, *rating* et *starring*, toutes de type **String**, sauf la colonne *rating*, de type **Double**.

   Cliquez sur **OK** pour fermer l'éditeur.

4. Dans la boîte de dialogue **[Propagate]**, cliquez sur **Yes** pour propager le schéma aux composants suivants.

5. Dans le champ **Filename**, saisissez le chemin d'accès au fichier JSON.

   Dans cet exemple, le fichier JSON se présente comme suit :

```
{"movieCollection": [
    {
        "type": "Action Movie",
        "name": "Brave Heart",
        "details": {
```

```
            "release": "1995",
            "rating": "5",
            "starring": "Mel Gibson"
        }
    },
    {
        "type": "Action Movie",
        "name": "Edge of Darkness",
        "details": {
            "release": "2010",
            "rating": "5",
            "starring": "Mel Gibson"
        }
    }
]}
```

6. Assurez-vous que la case **Read By XPath** est bien cochée.

7. Dans le champ **Loop JSONPath query**, saisissez *"/movieCollection/details"*.

8. Dans la table **Mapping**, le schéma apparaît automatiquement dans la colonne **Column**.

Mapping

Column	XPath query	Get Nodes
type	"../type"	
name	"../name"	
release	"release"	
rating	"rating"	
starring	"starring"	

Dans la colonne **XPath query**, saisissez les requêtes suivantes :

- Dans les colonnes *type* et *name*, saisissez les requêtes XPath *"../type"* et *"../name"*, respectivement. Elles correspondent aux premiers nœuds des données JSON.
- Pour les colonnes *release*, *rating* et *starring*, saisissez les requêtes XPath *"release"*, *"rating"* et *"starring"*, respectivement.

9. Double-cliquez sur le **tLogRow** pour afficher la vue **Basic settings** et sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter le Job.

```
[statistics] connecting to socket on port 3788
[statistics] connected
.------------+----------------+-------+------+----------.
|                      show_data                          |
|=-----------+----------------+-------+------+---------=|
|type        |name            |release|rating|starring  |
|=-----------+----------------+-------+------+---------=|
|Action Movie|Brave Heart     |1995   |5.0   |Mel Gibson|
|Action Movie|Edge of Darkness|2010   |5.0   |Mel Gibson|
'------------+----------------+-------+------+----------'
[statistics] disconnected
```

Comme affiché ci-dessus, les données source JSON sont collectées dans une table de fichier plat.

Scénario 3 : Extraire des données JSON à partir d'une URL

Dans ce scénario, le **tFileInputJSON** récupère le nœud *friends* à partir d'un fichier JSON contenant les données d'un utilisateur Facebook et le **tExtractJSONFields** extrait les données à partir du nœud *friends* pour une sortie contenant des données plates.

Notez que le fichier JSON est déployé sur le serveur Tomcat dans le dossier *<tomcat path>/webapps/docs*.

Relier les composants

1. Déposez un **tFileInputJSON**, un **tExtractJSONFields** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.

2. Reliez les composants **tFileInputJSON** et **tExtractJSONFields** entre eux à l'aide d'un lien **Row** > **Main**.

3. Reliez les composants **tExtractJSONFields** et **tLogRow** entre eux à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFileInputJSON** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur de schéma.

Cliquez sur le bouton **[+]** afin d'ajouter une colonne, *friends*, de type **String**.

Cliquez sur **OK** pour fermer l'éditeur.

3. Décochez la case **Read by XPath** et cochez la case **Use Url**.

Dans le champ **URL**, saisissez l'URL du fichier JSON, *"http://localhost:8080/docs/facebook.json"* dans ce scénario.

Dans cet exemple, le fichier JSON se présente comme suit :

```
{ "user": { "id": "9999912398",
            "name": "Kelly Clarkson",
            "friends": [
                 { "name": "Tom Cruise",
                   "id": "55555555555555",
                   "likes": {
                       "data": [
                            { "category": "Movie",
                              "name": "The Shawshank Redemption",
                              "id": "103636093053996",
                              "created_time": "2012-11-20T15:52:07+0000"
                            },
                            { "category": "Community",
```

```
                            "name": "Positiveretribution",
                            "id": "471389562899413",
                            "created_time": "2012-12-16T21:13:26+0000"
                          }
                              ]
                          }
                  },
                  { "name": "Tom Hanks",
                    "id": "88888888888888"
                    "likes": {
                         "data": [
                          { "category": "Journalist",
                            "name": "Janelle Wang",
                            "id": "136009823148851",
                            "created_time": "2013-01-01T08:22:17+0000"
                          },
                          { "category": "Tv show",
                            "name": "Now With Alex Wagner",
                            "id": "305948749433410",
                            "created_time": "2012-11-20T06:14:10+0000"
                          }
                          ]
                         }
                   }
                         ]
            }
}
```


4. Saisissez l'URL dans votre navigateur. Si le serveur Tomcat est opérationnel, le navigateur affiche :

```
{ "user": { "id": "9999912398",
            "name": "Kelly Clarkson",
            "friends": [
                 { "name": "Natalie Paulson",
                   "id": "33333333333333",
                   "likes": {
                       "data": [
                            { "category": "Movie",
                              "name": "The Shawshank Redemption",
                              "id": "103636093053996",
                              "created_time": "2012-11-20T15:52:07+0000"
                            },
                            { "category": "Community",
                              "name": "Positiveretribution",
                              "id": "471389562899413",
                              "created_time": "2012-12-16T21:13:26+0000"
                            }
                                ]
                            }
                 },
                 { "name": "Stephanie Chumley",
                   "id": "88888888888888"
```


5. De retour dans le studio, dans la table **Mapping**, saisissez la requête JSONPath *"$.user.friends[*]"* à côté de la colonne *friends*. Le nœud *friends* est récupéré à partir du fichier source.

6. Double-cliquez sur le **tExtractJSONFields** pour ouvrir sa vue **Basic settings**.

7. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur de schéma.

8. Cliquez cinq fois sur le bouton **[+]** dans le panneau de droite pour ajouter cinq colonnes, respectivement *id*, *name*, *like_id*, *like_name* et *like_category*. Ces colonnes contiendront les données des nœuds du champ JSON *fields*.

   Cliquez sur **OK** pour fermer l'éditeur.

9. Dans la boîte de dialogue **[Propagate]**, cliquez sur **Yes** pour propager le schéma aux composants suivants.

10. Dans le champ **Loop XPath query**, saisissez *"/likes/data"*.

11. Dans la table **Mapping**, saisissez les nœuds JSON dans la colonne **XPath query**. Les données de ces nœuds sont extraites et passées aux colonnes équivalentes définies dans le schéma de sortie.

12. Configurez la requête XPath *"../../id"* (interrogeant le nœud *"/friends/id"*) pour la colonne *id*.

Configurez également la requête *"../../name"* (interrogeant le nœud *"/friends/name"*) pour la colonne *name*, ainsi que les requêtes *"id"* pour la colonne *like_id*, *"name"* pour la colonne *like_name* et *"category"* pour la colonne *like_category*.

13. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.

Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.

2. Cliquez sur **F6** pour exécuter le Job.

```
.---------------+-----------------+---------------+-------------------------+-------------.
|                                          tLogRow_1                                        |
|=--------------+-----------------+---------------+-------------------------+------------=|
|id             |name             |like_id        |like_name                |like_category|
|=--------------+-----------------+---------------+-------------------------+------------=|
|33333333333333|Natalie Paulson   |103636093053996|The Shawshank Redemption|Movie         |
|33333333333333|Natalie Paulson   |471389562899413|Positiveretribution     |Community     |
|88888888888888|Stephanie Chumley|136009823148851|Janelle Wang            |Journalist    |
|88888888888888|Stephanie Chumley|305948749433410|Now With Alex Wagner    |Tv show       |
'---------------+-----------------+---------------+-------------------------+-------------'
```

Comme affiché ci-dessus, les données relatives aux amis de l'utilisateur Facebook Kelly Clarkson sont correctement extraites.

tFileInputLDIF

Propriétés du tFileInputLDIF

Famille du composant	File/Input	
Fonction	Le composant **tFileInputLDIF** lit un fichier LDIF donné ligne par ligne.	
Objectif	Le **tFileInputLDIF** ouvre un fichier qu'il lit ligne par ligne et passe les lignes complètes au composant suivant comme défini dans le schéma, via une connexion de type Row.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Chemin d'accès et nom du fichier, et/ou variable à traiter. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	add operation as prefix when the entry is modify type	Cochez cette case pour afficher le mode d'opérations effectuées.
	Value separator	Saisissez le séparateur requis pour découper les données dans un fichier donné. Par défaut, le séparateur utilisé est " ,".
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
Advanced settings	*Encoding*	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

		Le type de données des colonnes qui sera traité par cette option est **byte[]** et vous devrez la définir dans l'éditeur du schéma d'entrée.
	Use field options (for Base64 decode checked)	Cochez cette case afin de spécifier les colonnes encodées en Base64 dans le flux d'entrée. Une fois cochée, cette case active la table **Decode Base64 encoding values**, vous permettant ainsi de préciser les colonnes à décoder de Base64.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour lire des lignes complètes dans un fichier délimité pouvant être très gros. Ce composant permet de créer un flux de données à l'aide d'un lien **Row** > **Main**, ainsi que de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d'utilisation de ces deux liens, consultez *Scénario 2 : Extraire les données XML erronées dans un flux de rejet* du composant **tFileInputXML**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Ecrire des données dans un fichier LDIF*.

tFileInputMail

Propriétés du tFileInputMail

Famille de composant	File/Input	
Fonction	Le composant **tFileInputMail** lit l'en-tête et le contenu d'un fichier e-mail MIME ou MSG défini.	
Objectif	Ce composant permet d'extraire les données standard d'un fichier e-mail.	
Basic settings	*File Name*	Spécifiez le fichier e-mail dont vous souhaitez lire et extraire les données.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Mail type	Sélectionnez un type d'e-mail dans la liste déroulante : **MIME**ou **MSG**.
	Attachment export directory	Spécifiez le répertoire dans lequel vous souhaitez exporter les pièces jointes.
	Mail parts	Spécifiez pour chaque colonne de schéma les données à extraire du fichier e-mail MIME défini. • **Column** : Les cellules **Column** sont automatiquement renseignées avec le nom de colonne défini dans le schéma que vous avez propagé. • **Mail part** : Saisissez l'étiquette d'en-tête ou de corps de message à extraire de l'e-mail défini. • **Multi value** : Cochez cette case pour chaque colonne comportant des champs à valeurs multiples. • **Separator** : Saisissez le séparateur de valeur des champs à valeurs multiples.

		Cette table apparaît uniquement lorsque **MIME** est sélectionné dans la liste déroulante **Mail type**.
	MSG Mail parts	Spécifiez pour chaque colonne de schéma les données à extraire du fichier e-mail MSG défini. • **Column** : Les cellules **Column** sont automatiquement renseignées avec le nom de colonne défini dans le schéma que vous avez propagé. • **Mail part** : Cliquez sur chaque cellule puis sélectionnez une partie d'e-mail à extraire. Cette table apparaît uniquement lorsque **MSG** est sélectionné dans la liste déroulante **Mail type**.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour la ligne en erreur et terminer le processus avec les lignes sans erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**EXPORTED_FILE_PATH** : répertoire dans lequel exporter la pièce jointe de l'e-mail. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant traite un flux de données et par conséquent requiert un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Extraire des champs clés d'un e-mail

Ce scénario est constitué de deux composants et a pour objectif d'extraire certains champs clés et d'en afficher les valeurs dans la console **Run**.

1. Cliquez et déposez les composants **tFileInputMail** et **tLogRow**.
2. Reliez les composants à l'aide d'un lien de type **Row Main**.
3. Double-cliquez sur le composant **tFileInputMail** pour afficher l'onglet **Basic settings** de la vue **Component** et configurer les paramètres de base :

4. Dans le champ **File Name**, sélectionnez le fichier e-mail à traiter. Renseignez manuellement le chemin d'accès ou cliquez sur le bouton **[...]** pour parcourir vos dossiers jusqu'au fichier choisi.

5. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir la boîte de dialogue permettant de définir le schéma à afficher en sortie.

6. Définissez votre schéma en incluant toutes les colonnes que vous souhaitez retrouver dans le flux de sortie. Dans cet exemple, le schéma comporte quatre colonnes *Date*, *Author*, *Object* et *Status*.

7. Une fois défini, cliquez sur **OK** pour fermer la boîte de dialogue et propager le schéma dans le tableau **Mail parts**.

8. Dans la colonne **Mail part** du tableau, saisissez les clés standards des parties header (en -tête) et body (corps de message) qui seront utilisées pour retrouver les valeurs à envoyer en sortie.

9. Cochez la case **Multi Value** pour chacune des clés standards pouvant se retrouver plus d'une fois dans votre fichier d'entrée. Définissez si besoin un séparateur pour les valeurs de ces clés dans le champ **Separator**.

10. Double-cliquez sur le composant **tLogRow** pour afficher l'onglet **Basic settings** de la vue **Component** et configurer les paramètres de base.

11. Définissez le composant **tLogRow** afin que les valeurs affichées soient séparées par un retour chariot. Sous Windows, saisissez *\n* entre guillemets doubles. Pour plus d'informations sur le composant **tLogRow**, consultez *tLogRow*.

12. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

13. Appuyez sur **F6** our sur cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter et afficher le flux de sortie dans la console de la vue **Run Job**.

```
Starting job tFileInputMail at 11:48 06/11/2009.

[statistics] connecting to socket on port 3437
[statistics] connected
Wed, 4 Nov 2009 19:12:47 +0800
(CST)|musicatcher@gmail.com|Talend multi value test|by
10.142.186.14 with SMTP id j14cs69293wff;
        Wed, 4 Nov 2009 03:13:00 -0800 (PST)by 10.150.45.40
with SMTP id s40mr2413104ybs.260.1257333179981;
        Wed, 04 Nov 2009 03:12:59 -0800 (PST)from
mail-gx0-f210.google.com (mail-gx0-f210.google.com
[209.85.217.210])
        by mx.google.com with ESMTP id
33si1593716ywh.127.2009.11.04.03.12.58;
        Wed, 04 Nov 2009 03:12:58 -0800 (PST)by
mail-gx0-f210.google.com with SMTP id 2so6057662gxk.4
        for <musicatcher0@gmail.com>; Wed, 04 Nov 2009 03:12:58
-0800 (PST)by 10.150.75.12 with SMTP id
x12mr2370266yba.341.1257333175484;
        Wed, 04 Nov 2009 03:12:55 -0800 (PST)from nsun
([219.237.242.224])
        by mx.google.com with ESMTPS id
5sm364705ywd.23.2009.11.04.03.12.53
        (version=SSLv3 cipher=RC4-MD5);
        Wed, 04 Nov 2009 03:12:54 -0800 (PST)
[statistics] disconnected
Job tFileInputMail ended at 11:48 06/11/2009. [exit code=0]
```

Les valeurs clés d'en-tête sont extraites comme spécifié dans le tableau **Mail parts**. En effet, la date de réception, l'auteur, le sujet ainsi que les différents statuts contenus dans l'e-mail sont affichés.

tFileInputMSDelimited

Propriétés du tFileInputMSDelimited

Famille du composant	File/Input	
Fonction	Le composant **tFileInputMSDelimited** permet de lire des fichiers délimités multi-structure complexes.	
Objectif	Le **tFileInputMSDelimited** ouvre le fichier multi structure complexe, en lit la structure (schémas) puis utilise des liens de type **Row** pour envoyer les champs, tels qu'ils ont été définis dans les différents schémas, vers les composants suivants dans le Job.	
Basic settings	*Multi Schema Editor*	L'éditeur de schémas **[Multi Schema Editor]** permet de construire et de configurer le flux de données dans un fichier délimité multi-structure pour pouvoir associer un schéma pour chaque sortie. Pour plus d'informations, consultez *L'éditeur de schémas Multi Schema Editor*.
	Output	Les schémas définis dans l'éditeur de schémas **[Multi Schema Editor]**, ainsi que les types d'enregistrements correspondants, s'affichent automatiquement dans le tableau **Output**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur.
Advanced settings	*Trim all columns*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes.
	Validate date	Cochez cette case pour vérifier strictement le format de la date par rapport au schéma d'entrée.
	Advanced separator (for numbers)	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour lire des fichiers délimités multi structure et définissez un séparateur de champs pour différencier les champs contenus dans ces fichiers.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

L'éditeur de schémas Multi Schema Editor

L'éditeur **[Multi Schema Editor]** vous permet de définir :

- le chemin d'accès au fichier source,
- les propriétés du fichier source,
- la structure des données pour chacun des schémas de sortie.

Lorsque vous définissez la structure de chaque schéma de sortie dans le **[Multi Schema Editor]**, le nom des colonnes des différentes structures s'affichent automatiquement dans la liste du schéma d'entrée des composants qui suivent le **tFileInputMSDelimited**. Vous pouvez cependant définir la structure des données directement dans la vue **Basic settings** de chacun de ces composants.

Le **[Multi Schema Editor]** permet également de sélectionner un schéma qui servira de source (clé primaire) aux données entrantes, afin d'en assurer l'unicité. L'éditeur utilise ce procédé de mapping pour associer tous les schémas traités dans le fichier délimité au schéma source de ce même fichier.

L'éditeur s'ouvre en affichant par défaut la première colonne. Cette colonne contient généralement l'indicateur de type des enregistrements. Une fois ouvert, vous pouvez cependant définir comme clé primaire n'importe quelle colonne du schéma en cochant la case associée.

L'image ci-dessous donne un aperçu de l'éditeur de schémas **[Multi Schema Editor]**.

Pour des informations plus détaillées sur le **Multi Schema Editor**, consultez *Scénario : Lecture d'un fichier délimité multi-structure*.

Scénario : Lecture d'un fichier délimité multi-structure

Le présent scénario sert à créer un Job en Java permettant de lire trois schémas dans un même fichier délimité et d'en afficher la structure sur la console de la vue **Run**.

Le fichier délimité traité dans ce scénario se présente comme suit :

```
01;SOFT MUSIC ALBUM;RICHARDSON;15/12/2005
02;We Danced
02;She's Everything
02;Once in a Lifetime Love
03;National Library
01;COUNTRY MUSIC ALBUM;WHITE;02/01/2006
02;Fall Into Me
02;Another Try
02;Something About Her
```

Déposer et relier les composants

1. A partir de la **Palette**, déposez les composants suivants dans l'espace de modélisation graphique : un composant **tFileInputMSDelimited** et trois **tLogRow**.

2. Dans l'espace de modélisation graphique, faites un clic-droit sur le composant **tFileInputMSDelimited** et connectez-le aux trois composants **tLogRow** (nommés ici **tLogRow1**, **tLogRow2** et **tLogRow3** à l'aide de liens **row** nommés respectivement **row_A_1**, **row_B_1** et **row_C_1**.

Configurer les composants

1. Double-cliquez sur le **tFileInputMSDelimited** pour ouvrir l'éditeur **Multi Schema Editor**.

2. Cliquez sur le bouton **Browse...** à côté du champ **File name** pour indiquer où se trouve le fichier délimité multi-schéma que vous voulez traiter.

3. Dans la zone **File Settings**, renseignez :

   -le type d'encodage de votre fichier source, à partir de la liste déroulante. Cette option permet de garantir une certaine cohérence entre tous les fichiers d'entrée et de sortie.

   -les séparateurs de champ et de ligne utilisés dans le fichier source.

   Cochez la case **Use Multiple Separator** et configurez les champs qui suivent si différents séparateurs de champs sont utilisés pour séparer les schémas du fichier source.

   Un aperçu des données du fichier source apparaît automatiquement dans le panneau d'affichage **Preview**.

La colonne **Column 0**, qui contient généralement l'indicateur de type des enregistrements est sélectionnée par défaut. Vous pouvez cependant définir n'importe quelle autre colonne comme clé primaire en cochant sa case associée.

4. Cliquez sur le bouton **Fetch Codes** à droite du panneau **Preview** pour cataloguer les types de schémas et les enregistrements du fichier source. Dans ce scénario, le fichier source comporte trois types de schémas (*A*, *B*, *C*).

   A partir du panneau d'affichage **Fetch Codes**, cliquez sur chacun des trois types pour que la structure de leurs données s'affiche dans le panneau **Preview**.

5. Cliquez dans les cellules de la ligne *Name* et choisissez le nom des colonnes de chacun des schémas sélectionnés.

   Dans ce scénario, les colonnes se lisent comme suit :

   -Schéma *A* : *Type*, *DiscName*, *Author*, *Date*,

   -Schéma *B* : *Type*, *SongName*,

-Schéma *C* : *Type*, *LibraryName*.

Vous devez à présent définir, à partir des données d'entrée, la clé primaire qui permettra de conserver leur unicité (dans ce scénario, il s'agit de *DiscName*). Il vous faut pour cela :

6. Sélectionner, à partir du panneau d'affichage **Fetch Codes**, le schéma contenant la colonne que vous voulez définir comme clé primaire (dans ce scénario, il s'agit du schéma *A*) pour en afficher la structure.

7. Cliquer dans la cellule *Key* correspondant à la colonne *DiscName* et cocher la case qui s'affiche alors.

8. Cliquer n'importe où dans l'éditeur pour que la mention *false*, dans la case *Key*, devienne *true*.

Vous devez maintenant indiquer le schéma "père" auquel vous voulez rattacher les autres schémas "fils" (dans ce scénario, il s'agit de *DiscName*). Il vous faut pour cela :

9. Sélectionner le schéma *B*, à partir du panneau d'affichage **Fetch Codes**, puis cliquer sur le bouton flèche droite pour le déplacer vers la droite.

Faire de même pour le schéma *C*.

Le champ **Cardinality** n'est pas obligatoire. Il vous permet de déterminer le nombre (ou l'intervalle) de champs des schémas "fils" rattachés au schéma "père". Cependant, si vous entrez un nombre ou un intervalle inexact et essayez d'exécuter le Job, un message d'erreur apparaîtra.

10. Dans le **[Multi Schema Editor]**, cliquez sur le bouton **OK** pour valider tous les changements que vous venez d'effectuer et fermer l'éditeur de schémas.

Les trois schémas définis, ainsi que les types d'enregistrements correspondants, s'affichent automatiquement dans la vue **Basic settings** du composant **tFileInputMSDelimited**.

Les trois schémas que vous avez définis dans l'éditeur **[Multi Schema Editor]** sont automatiquement communiqués aux trois composants **tLogRow**.

11. Si nécessaire, cliquez sur le bouton **Edit schema** dans la vue **Basic settings** de chacun des composants **tLogRow** pour avoir un aperçu des structures de données en entrée et en sortie, telles que vous les avez définies dans le **Multi Schema Editor**, ou pour les modifier.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

   Le fichier délimité multi-schéma est lu ligne par ligne et les champs extraits sont affichés dans la console **Run Job** comme défini dans le **[Multi Schema Editor]**.

```
Starting job pivot at 10:33 18/01/2010.

01|SOFT MUSIC ALBUM|RICHARDSON|15/12/2005
02|We Danced|SOFT MUSIC ALBUM
02|She's Everything|SOFT MUSIC ALBUM
02|Once in a Lifetime Love|SOFT MUSIC ALBUM
03|National Library|SOFT MUSIC ALBUM
01|COUNTRY MUSIC ALBUM|WHITE|02/01/2006
02|Fall Into Me|COUNTRY MUSIC ALBUM
02|Another Try|COUNTRY MUSIC ALBUM
02|Something About Her|COUNTRY MUSIC ALBUM
Job pivot ended at 10:33 18/01/2010. [exit code=0]
```

tFileInputMSPositional

Propriétés du tFileInputMSPositional

Famille du composant	File/Input	
Fonction	Le composant **tFileInputMSPositional** permet de lire des schémas multiples à partir d'un fichier positionnel.	
Objectif	Le **tFileInputMSPositional** ouvre le fichier multi structure complexe, en lit la structure (schémas) puis utilise des liens de type **Row** pour envoyer les champs, tels qu'ils ont été définis dans les différents schémas, vers les composants suivants dans le Job.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File Name	Chemin d'accès et nom du fichier, et/ou variable à traiter. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Row separator	Chaîne de caractères (ex : "\n" sous Unix) séparant les lignes.
	Header Field Position	Position de la chaîne de caractères identifiant les différents schémas.
	Records	**Schema** : définissez autant de schémas que nécessaire. **Header value** : valeur de la chaîne de caractères identifiant les différents schémas. **Pattern** : Chaîne de caractères représentant la longueur de chacune des colonnes du schéma correspondant, séparées par des virgules. Vérifiez que les valeurs définies dans ce champ sont cohérentes avec le schéma défini. **Reject incorrect row size** : Cochez les cases des schémas pour lesquels vous n'acceptez pas une taille de ligne incorrecte. **Parent row**: Sélectionnez la ligne parent dans la liste déroulante. Par défaut, il s'agit d'*<Empty>*. **Parent key column** : Saisissez le nom de la colonne clé parente. Si la ligne parent n'est pas *<Empty>*, vous devez saisir le nom de la colonne de la ligne parent du schéma dans ce champ. **Key column** : Saisissez le nom de la colonne clé.
	Skip from header	Nombre de lignes à ignorer au début du fichier.
	Skip from footer	Nombre de lignes à ignorer à la fin du fichier.
	Limit	Nombre maximal de lignes à traiter. Si Limit = 0, aucune ligne ne sera lue ni traitée.
	Die on parse error	Décochez cette case pour passer les lignes contenant des erreurs de parse et terminer le traitement avec les lignes sans erreur.
	Die on unknown header type	Décochez cette case pour passer les lignes dont le type d'en-tête est inconnu et terminer le traitement avec les lignes sans erreurs.

Advanced settings	*Process long rows (needed for processing rows longger than 100,000 characters wide)*	Cochez cette case afin de pouvoir traiter de longues lignes (ceci est nécessaire pour traiter des lignes contenant plus de 100 000 caractères).
	Advanced separator (for numbers)	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Trim all columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes.
	Validate date	Cochez cette case pour vérifier strictement le format de la date par rapport au schéma d'entrée.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **Flow** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **Flow** et retourne un nombre entier. **NB_LINE_UNKOWN_HEADER_TYPES** : nombre de lignes ayant un type d'en-tête inconnu. Cette variable est une variable **Flow** et retourne un nombre entier. **NB_LINE_PARSE_ERRORS** : nombre de lignes ayant des erreurs de parsage. Cette variable est une variable **Flow** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour lire un fichier positionnel multi-schéma et servez-vous d'une valeur spécifiant l'emplacement pour séparer les champs. Ce composant permet de créer un flux de données à l'aide d'un lien **Row** > **Main**, ainsi que de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d'utilisation de ces deux liens, consultez *Scénario 2 : Extraire les données XML erronées dans un flux de rejet* du composant **tFileInputXML**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Lire des données d'un fichier positionnel

Le scénario suivant permet de lire des données d'un fichier positionnel contenant deux schémas. Le fichier positionnel se présente comme suit :

```
schema_1 (car_owner):schema_id;car_make;owner;age
```

```
schema_2 (car-insurance):schema_id;car_owner;age;car_insurance
1bmw     John      45
1bench   Mike      30
2John      45 yes
2Mike      50 No
```

Déposer les composants

1. Déposez un **tFileInputMSPositional** et deux composants **tLogRow** de la **Palette** dans l'espace de modélisation graphique.

2. Renommez les deux composants **tLogRow**, respectivement **car_owner** et **car_insurance**.

Configurer les composants

1. Double-cliquez sur le **tFileInputMSPositional** pour afficher sa vue **Basic settings** et paramétrer ses propriétés.

2. Dans le champ **File name/Stream**, saisissez le chemin d'accès à votre fichier d'entrée. Vous pouvez également cliquer sur le bouton **[...]** pour parcourir votre système et sélectionner votre fichier.

3. Dans le champ **Header Field Position**, saisissez la position de début et de fin de l'en-tête pour l'identifiant du schéma dans le fichier d'entrée, *0-1*, dans ce cas, puisque le premier caractère de chaque ligne est l'identifiant du schéma.

4. Cliquez deux fois sur le bouton **[+]** pour ajouter deux lignes à la table **Records**.

5. Cliquez sur la cellule dans la colonne **Schema** pour afficher le bouton **[...]**.

   Cliquez sur le bouton **[...]** pour ouvrir la boîte de dialogue de nommage du schéma.

6. Saisissez le nom du schéma et cliquez sur **OK**.

   Le nom du schéma apparaît dans la cellule et l'éditeur du schéma s'ouvre.

7. Définissez le schéma *car_owner*, contenant quatre colonnes : *schema_id*, *car_make*, *owner* et *age*.

8. Répétez ces étapes pour définir le schéma *car_insurance*, contenant quatre colonnes : *schema_id*, *car_owner*, *age* et *car_insurance*.

9. Connectez le composant **tFileInputMSPositional** au **car_owner** à l'aide d'un lien **Row** > **car_owner** et au composant **car_insurance** à l'aide d'un lien **Row** > **car_insurance**.

10. Dans la colonne **Header value**, saisissez l'identifiant du schéma, *1* pour le schéma *car_owner* et *2* pour le schéma *car_insurance*.

11. Dans la colonne **Pattern**, saisissez la longueur de chaque champ du schéma, c'est-à-dire le nombre de caractères, *1,8,10,3* pour le schéma *car_owner* et *1,10,3,3* pour le schéma *car_insurance* dans ce scénario.

12. Dans le champ **skip from header**, saisissez le nombre de lignes d'en-tête à ignorer, *2* dans cet exemple, car les deux premières lignes décrivent les deux schémas et ne fournissent aucune valeur.

13. Sélectionnez **Table (print values in cells of a table)** dans la zone **Mode** des composants **car_owner** et **car_insurance**.

Enregistrer et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

```
[statistics] connecting to socket on port 3754
[statistics] connected
.---------+---------+---+-------------.
|            car_insurance             |
|=--------+---------+---+------------=|
|schema_id|car_owner|age|car_insurance|
|=--------+---------+---+------------=|
|2        |John     |45 |yes          |
|2        |Mike     |50 |No           |
'---------+---------+---+-------------'

.---------+--------+-----+---.
|         car_owner          |
|=--------+--------+-----+--=|
|shcema_id|car_make|owner|age|
|=--------+--------+-----+--=|
|1        |bmw     |John |45 |
|1        |bench   |Mike |30 |
'---------+--------+-----+---'
```

Le fichier est lu ligne par ligne selon les valeurs de longueur définies dans le champ **Pattern** et écrit en sortie dans deux tables ayant deux schémas différents.

tFileInputMSXML

Propriétés du tFileInputMSXML

Famille du composant	XML ou File/Input	
Fonction	Le composant **tFileInputMSXML** permet de lire des schémas multiples à partir d'un fichier XML structuré et de les restituer en sortie.	
Objectif	Le **tFileInputMSXML** ouvre le fichier multi structure complexe, en lit la structure (schémas) puis utilise des liens de type **Row** pour envoyer les champs, tels qu'ils ont été définis dans les différents schémas, vers les composants suivants dans le Job.	
Basic settings	*File Name*	Chemin d'accès et nom du fichier, et/ou variable à traiter. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Root XPath query	Racine de l'arborescence XML sur laquelle se base la requête.
	Enable XPath in column "Schema XPath loop" but lose the order	Cochez cette case si vous souhaitez définir un chemin XPath dans le champ **Schema XPath loop** du tableau **Outputs** sans garder l'ordre des données montrées dans le fichier XML source. *Cette option ne prend effet que si vous sélectionnez le mode de génération **dom4j** dans la vue **Advanced settings**.*
	Outputs	**Schema** : Définissez autant de schémas que nécessaire. **Schema XPath loop** : Saisissez le nœud ou le chemin XPath de l'arborescence XML sur lequel la boucle est basée. **XPath Queries** : Renseignez les champs à extraire de la structure XML d'entrée. **Create empty row** : Cochez cette case si vous souhaitez créer des lignes vides pour le(s) champ(s) vide(s) du schéma.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur.
Advanced settings	*Trim all column*	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes.
	Validate date	Cochez cette case pour vérifier strictement le format de la date par rapport au schéma d'entrée.
	Ignore DTD file	Cochez cette case afin d'ignorer le fichier DTD indiqué dans le fichier XML traité.
	Generation mode	Sélectionnez le mode de génération correspondant à votre mémoire disponible. Les modes disponibles sont : • Lent et consommateur de mémoire (**Slow and memory-consuming - Dom4J**). Cette option vous permet d'utiliser Dom4J pour traiter des fichiers XML très complexes. • Rapide et peu consommateur de mémoire (**Fast with low memory consumption (SAX)**).
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Lecture d'un fichier XML multi-structuré

Le présent scénario décrit un Job permettant de lire un fichier XML multi-structuré, d'en extraire les champs désirés et de les afficher dans la console.

Construire le Job

1. Déposez un **tFileInputMSXML** de la **Palette** dans l'espace de modélisation graphique et double-cliquez sur le composant afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Parcourez votre système jusqu'au fichier XML que vous souhaitez traiter, *D:/Input/multischema_xml.xml* dans cet exemple. Ce fichier qui contient les données suivantes :

```
<root>
        <toy>Cat</toy>
        <record>We Belong Together</record>
        <book>As You Like It</book>
        <book>All's Well That Ends Well</book>
        <record>When You Believe</record>
        <toy>Dog</toy>
</root>
```

3. Dans le champ **Root XPath query**, indiquez la racine de l'arborescence XML sur laquelle sera basée la requête, *"/root"* dans cet exemple.

4. Cochez la case **Enable XPath in column "Schema XPath loop" But lose the order**.

   Dans cet exemple, afin d'extraire les champs voulus, vous devez définir un chemin XPath dans le champ **Schema XPath loop** du tableau **Outputs** pour chaque flux de sortie sans que l'ordre des données montrées dans le fichier XML source ne soit gardé.

5. Cliquez sur le bouton **[+]** pour ajouter des lignes au tableau **Outputs** dans lesquelles vous pouvez définir les schémas de sortie, *record* et *book* dans cet exemple.

6. Dans le tableau **Outputs**, cliquez dans chaque cellule **Schema** et cliquez sur le bouton **[...]** pour saisir le nom du schéma dans la boîte de dialogue qui s'ouvre.

   Saisissez un nom pour le schéma de sortie et cliquez sur **OK** afin de fermer la boîte de dialogue.

7. L'éditeur de schéma du **tFileInputMSXML** s'affiche.

   Définissez le schéma selon vos besoins.

8. Effectuez cette procédure afin de définir le schéma de sortie *record*.

9. Dans la cellule **Schema XPath loop**, indiquez le nœud de l'arborescence XML sur laquelle sera basée la requête. Dans cet exemple, saisissez *"/book"* et *"/record"* respectivement.

10. Dans la cellule **XPath Queries**, indiquez les champs à extraire du fichier XML d'entrée. Dans cet exemple, saisissez la requête XPath *"."*.

11. Dans l'espace de modélisation graphique, déposez deux **tLogRow** de la **Palette** et reliez le **tFileInputMSXML** au **tLogRow1** et au **tLogRow2** à l'aide des liens **book** et **record** respectivement.

Renommez les deux composants **tLogRow** *book* et *record* respectivement.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur **Run** dans l'onglet **Run** afin d'exécuter le Job.

   Le fichier XML multi-structuré est lu ligne par ligne et les champs extraits s'affichent dans la console. Les deux premiers champs sont pour le schéma *book* et les deux autres pour le schéma *record*.

```
Starting job check at 17:58 07/05/2014.

[statistics] connecting to socket on port 3985
[statistics] connected
As You Like It
All's Well That Ends Well
We Belong Together
When You Believe
[statistics] disconnected
Job check ended at 17:58 07/05/2014. [exit code
```

tFileInputPositional

Propriétés du tFileInputPositional

Famille de composant	File/Input	
Fonction	Le composant **tFileInputPositional** lit un fichier ou un flux de données ligne par ligne et extrait les champs selon un modèle (*pattern*). Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job Map/Reduce **Talend** afin de générer du code Map/Reduce. Pour plus d'informations, consultez *Le tFileInputPositional dans des Jobs Map/Reduce Talend*.	
Objectif	Le **tFileInputPositional** lit un fichier ou un flux de données ligne par ligne, sépare les champs tels que définis par le schéma et passe les données extraites au composant suivant via une connexion de type Row.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File name/Stream	**File name** : Chemin d'accès et nom du fichier à traiter. **Stream** : Flux de données à traiter. Les données doivent préalablement être mises en flux afin d'être récupérées par le **tFileInputPositional** via la variable représentative correspondante. Cette variable peut être prédéfinie dans votre Studio ou fournie par le contexte ou les composants utilisé(s) avec ce composant, par exemple la variable *INPUT_STREAM* du **tFileFetch**. Sinon, vous pouvez la définir manuellement et l'utiliser selon votre Job, par exemple à l'aide d'un **tJava** ou d'un **tJavaFlex**. Afin d'éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d'autocomplétion (**Ctrl +Espace**) afin de remplir le champ, si cette variable a été correctement définie. Pour plus d'informations concernant les variables disponibles, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations concernant les flux d'entrée, consultez *Scénario 2 : Lire les données d'un fichier distant en mode stream*.
	Row separator	Saisissez le séparateur à utiliser pour identifier la fin des lignes.
	Use byte length as the cardinality	Cochez cette case pour permettre la prise en charge des caractères à deux octets dans ce composant. Pour cette fonction, JDK 1.6 est requis.
	Customize	Cochez cette case pour personnaliser le format des données du fichier positionnel et renseignez les colonnes du tableau **Formats**. **Column** : Sélectionnez la colonne que vous souhaitez personnaliser. **Size** : Saisissez la taille correspondant à la colonne. **Padding char** : saisissez, entre guillemets, le caractère de remplissage à supprimer du champ. Le caractère par défaut est un espace. **Alignment** : Sélectionnez le paramètre d'alignement approprié.

	Pattern	Longueurs séparées par des virgules, interprétées comme une chaîne de caractères entre guillemets. Vérifiez que les valeurs saisies dans ce champ sont cohérentes avec le schéma défini.
	Skip empty rows	Cochez cette case pour ignorer les lignes vides.
	Uncompress as zip file	Cochez cette case pour décompresser le fichier d'entrée.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	Header	Saisissez le nombre de lignes à ignorer au début du fichier.
	Footer	Nombre de ligne à ignorer à la fin d'un fichier.
	Limit	Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n'est lue ou traitée.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Pour bénéficier de la fonctionnalité de schéma dynamique, vous pouvez utiliser ce composant avec le **tSetDynamicSchema**.
		Built-in : Le schéma sera créé et conservé pour ce composant. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*Needed to process rows longer than 100 000 characters*	Cochez cette case si les lignes à traiter dans le fichier d'entrée font plus de 100 000 caractères de long.
	Advanced separator (for numbers)	Cochez cette case pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : configurez le séparateurs des milliers. **Decimal separator** : configurez le séparateur pour les décimaux.
	Trim all columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes.
	Check columns to trim	Cochez la case devant le nom de chacune des colonnes dont vous souhaitez supprimer les espaces de début et de fin de champ.
	Validate date	Cochez cette case pour vérifier strictement le format de date par rapport au schéma d'entrée.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.

Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour lire un fichier et séparer les champs à l'aide du séparateur spécifié. Ce composant permet de créer un flux de données à l'aide d'un lien **Row** > **Main**, ainsi que de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d'utilisation de ces deux liens, consultez *Scénario 2 : Extraire les données XML erronées dans un flux de rejet* du composant **tFileInputXML**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Le tFileInputPositional dans des Jobs Map/Reduce Talend

*Les informations contenues dans cette section concernent uniquement les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend** et ne sont pas applicables aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job Map/Reduce **Talend**, le **tFileInputPositional**, ainsi que le Job complet utilisant ce composant, génère du code Map/Reduce natif. Cette section présente les propriétés spécifiques du **tFileInputPositional**lorsqu'il est utilisé dans un Job Map/Reduce. Pour plus d'informations concernant un Job Map/Reduce **Talend**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	MapReduce / Input	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Réutilisez les propriétés stockées centralement sous le nœud **Hadoop Cluster** de la vue **Repository**. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Pour plus d'informations concernant le nœud **Hadoop Cluster**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si vous effectuez des modifications, le schéma devient automatiquement built-in.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

	Folder/File	Parcourez votre système ou saisissez le chemin d'accès au répertoire de HDFS dans lequel sont stockées les données à utiliser. Si le chemin d'accès défini pointe vers un dossier, ce composant lit tous les fichiers stockés dans le dossier, par exemple */user/talend/in*. Si des sous-dossiers existent, ils sont automatiquement ignorés à moins que le chemin défini soit */user/talend/in/**. Si vous souhaitez spécifier plusieurs fichiers ou dossiers dans ce champ, séparez les chemins à l'aide d'une virgule (,). Si le fichier à lire est un fichier compressé, saisissez son nom ainsi que son extension. Le **tHDFSInput** décompresse le fichier automatiquement à l'exécution. Les formats de compression, ainsi que les extensions correspondantes, sont : • DEFLATE : **.deflate* • gzip : **.gz* • bzip2 : **.bz2* • LZO : **.lzo* Assurez-vous que la connexion à la distribution Hadoop utilisée est correctement configurée dans l'onglet **Hadoop configuration** de la vue **Run**.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	Row separator	Saisissez le séparateur à utiliser pour identifier la fin des lignes.
	Customize	Cochez cette case pour personnaliser le format de données du fichier positionnel et définissez les colonnes de la table : **Column** : sélectionnez la colonne à personnaliser. **Size** : saisissez la taille de la colonne. **Padding char** : saisissez, entre guillemets, le caractère de remplissage à supprimer du champ. Ce caractère est un espace par défaut. **Alignment** : sélectionnez le paramètre d'alignement approprié.
	Pattern	Saisissez entre guillemets les valeurs des longueurs séparées par une virgule, interprétées comme des chaînes de caractères. Assurez-vous que les valeurs saisies dans ce champ sont cohérentes par rapport au schéma défini.
	Header	Saisissez le nombre de lignes à ignorer au début du fichier. Saisissez *0* pour ne pas ignorer de ligne dans les données sans en-tête et *1* pour les données dont l'en-tête se trouve dans la première ligne.
	Skip empty rows	Cochez cette case pour ignorer les lignes vides.
Advanced settings	*Custom Encoding*	Il est possible de rencontrer des problèmes d'encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste **Encoding**. Sélectionnez l'encodage à utiliser dans la liste ou sélectionnez **Custom** pour le définir manuellement.
	Advanced separator (for number)	Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.).

	Trim columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées. Lorsque cette case est cochée, la table **Check column to trim** est affichée, ce qui vous permet de sélectionner les colonnes desquelles vous souhaitez enlever les espaces en début et en fin de champ.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme composant de début et requiert un composant de transformation comme lien de sortie. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Lorsqu'un Job Map/Reduce est ouvert dans l'espace de modélisation graphique, le **tFileInputDelimited** et la famille **MapReduce** s'affichent dans la **Palette** du Studio. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.	

Scénario : Transformer un fichier positionnel en fichier XML

Le scénario suivant construit un Job avec deux composants, qui a pour objectif de lire les données d'un fichier positionnel en entrée et de rendre des données sélectionnées en sortie (selon leur position) dans un fichier XML.

```
Contract        CustomerRef     InsuranceNr
00001           8200            50330
00001           8201            50331
00002           8202            50332
00002           8203            50333
```

Déposer et relier les composants

1. Cliquez et déposez un composant **tFileInputPositional** de la **Palette** dans l'espace de modélisation.
2. Cliquez-déposez un composant **tFileOutputXML**. Ce fichier recevra les références de manière structurée.

3. Cliquez-droit sur le composant **tFileInputPositional** et sélectionnez une connexion **Row** > **Main**. Glissez cette connexion vers le composant **tFileOutputXML** et relâchez la souris lorsque le symbole de prise de courant apparaît.

Configurer les données d'entrée

1. Sélectionnez le composant **tFileInputPositional** pour afficher sa vue **Basic settings** et définir ses propriétés.

2. Les propriétés de ce Job sont de type **Built-in** pour ce scénario.

   Par conséquent, les informations de propriétés sont renseignées pour ce Job seulement et ne peuvent être réutilisées pour un autre Job, contrairement à des propriétés de type **Repository**.

3. Renseignez le chemin d'accès au fichier dans le champ **File Name**. Ce champ est obligatoire.

4. Puis définissez le séparateur de lignes (**Row separator**) permettant d'identifier la fin de la ligne : le retour chariot par défaut.

5. Si nécessaire, cochez la case **Use byte length as the cardinality** pour permettre la prise en charge des caractères à deux octets.

6. Puis dans le champ **Pattern** définissant les champs d'une ligne. Le pattern est une série de longueurs correspondant aux valeurs de champs du fichier en entrée. Les valeurs doivent être saisies entre guillemets simples et séparées par une virgule. Veillez à ce que les valeurs saisies correspondent à la longueur des champs définis dans le schéma.

7. Renseignez les champs d'en-tête (**Header**), de pied de page (**Footer**) et de limite (**Limit**) selon la structure de votre fichier d'entrée et selon vos besoins. Dans ce scénario, ignorez la première ligne lors de la lecture du fichier d'entrée en saisissant *1* dans le champ **Header** et laissez les autres champs tels qu'ils sont.

8. Dans la liste **Schema**, sélectionnez **Repository** si le schéma d'entrée est stocké dans le Repository. Dans ce scénario, sélectionnez **Built-In** pour définir les données à transmettre au composant **tFileOutputXML**.

9. Vous pouvez sélectionner et/ou modifier le schéma via la fonction **Edit Schema**. Pour ce schéma, définissez trois colonnes, respectivement *Contracts*, *CustomerRef* et *InsuranceNr* correspondant aux trois valeurs de longueurs définies. Cliquez sur **OK** pour fermer la boîte de dialogue **[Schema]** et propager les modifications.

Configurer les données de sortie

1. Double-cliquez sur le composant **tFileOutputXML** afin d'afficher sa vue **Basic settings** et configurer ses propriétés de base.

2. Saisissez le chemin d'accès au fichier XML de sortie.

3. Définissez la balise de la ligne (**Row tag**) qui définit chaque ligne. Dans ce cas, la balise est *ContractRef*.

4. Cliquez sur le bouton **[+]** à côté du champ **Edit Schema** pour voir la structure de données, puis cliquez sur **Sync columns** pour récupérer la structure des données du composant d'entrée.

5. Cliquez sur l'onglet **Advanced settings** afin de configurer les paramètres avancées de la sortie XML.

6. Cliquez sur le bouton **[+]** pour ajouter une ligne dans la table **Root tags** et saisissez un nom de balise (ou plusieurs) pour encadrer la structure XML de sortie, *ContractsList* dans ce scénario.

7. Configurez les paramètres dans la table **Output format** si nécessaire. Par exemple, cochez la case **As attribute** d'une colonne si vous souhaitez utiliser son nom et sa valeur comme attribut pour l'élément XML parent. Décochez la case **Use schema colum name** pour que la colonne réutilise le libellé de la colonne d'entrée comme libellé de la balise. Dans ce scénario, laissez les paramètres par défaut.

8. Pour regrouper les lignes de sortie selon le numéro de contrat, cochez la case **Use dynamic grouping**, ajoutez une ligne dans la table **Group by**, sélectionnez **Contract** dans la liste **Column** et saisissez un attribut pour cette colonne dans le champ **Attribute label**.

9. Laissez les autres paramètres tels qu'ils sont.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job et vous assurer que tous les paramètres sont bien pris en compte.

2. Appuyez sur **F6** ou cliquez sur l'onglet **Run** puis sur le bouton **Run** pour exécuter le Job.

Le fichier est lu ligne par ligne selon les longueurs définies précédemment dans le champ **Pattern** et écrit en tant que fichier XML, comme défini dans les paramètres de sortie. Vous pouvez l'ouvrir dans n'importe quel éditeur XML standard.

```
1  <?xml version="1.0" encoding="ISO-8859-15"?>
2  <ContractsList>
3      <Contract Nr="00001">
4          <ContractRef>
5              <CustomerRef>8200</CustomerRef>
6              <InsuranceNr>50330</InsuranceNr>
7          </ContractRef>
8          <ContractRef>
9              <CustomerRef>8201</CustomerRef>
10             <InsuranceNr>50331</InsuranceNr>
11         </ContractRef>
12     </Contract>
13     <Contract Nr="00002">
14         <ContractRef>
15             <CustomerRef>8202</CustomerRef>
16             <InsuranceNr>50332</InsuranceNr>
17         </ContractRef>
18         <ContractRef>
19             <CustomerRef>8203</CustomerRef>
20             <InsuranceNr>50333</InsuranceNr>
21         </ContractRef>
22     </Contract>
23 </ContractsList>
```

tFileInputProperties

Propriétés du tFileInputProperties

Famille de composant	File/Input	
Fonction	Le composant **tFileInputProperties** lit un fichier texte ligne par ligne et extrait les champs.	
Objectif	Il ouvre un fichier texte qu'il lit ligne par ligne, sépare les champs sur le modèle clé=valeur.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Le schéma de ce composant est en lecture seule. Il se compose de deux colonnes, *Key* et *Value*, correspondant respectivement au nom du paramètre et à la valeur du paramètre à copier.
	File format	Sélectionnez le format de votre fichier **.properties** ou **.ini**.
		.properties : fichier de configuration dont les données sont disposées sur deux lignes et structurées de la manière suivante clé=valeur.
		.ini : fichier de configuration dont les données sont disposées sur deux lignes, structurées de la manière suivante clé=valeur et regroupées en sections. **Retrieve Mode** : - **Retrieve All** pour récupérer l'intégralité du fichier, - **Retrieve Sections** pour récupérer les sections du fichier, -**Retrieve by section** pour récupérer les données contenues dans la section spécifiée dans le champ **Section Name**. **Section Name** : saisissez le nom de la section sur laquelle effectuer l'opération d'itération.
	File	Chemin d'accès et nom du fichier, et/ou variable à traiter. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Calculate MD5 Hash	Cochez cette case pour que l'algorithme vérifie que le fichier est bien téléchargé.
Advanced settings	*Encoding*	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour lire un fichier et séparer les données en fonction de la structure clé=valeur du fichier.

Scénario : Lire et mapper la clé et les valeurs de fichiers properties et alimenter un glossaire

Le scénario suivant est un Job de quatre composants lisant deux fichiers propriétés en anglais et en français. Ces données sont ensuite mises en correspondance afin de constituer un glossaire.

Les fichiers de propriétés utilisés pour ce scénario sont des fichiers contenant les chaînes de caractères utilisés pour la localisation du composant **tMysqlInput** dans votre *Studio Talend*.

tMysqlInput_messages_en.properties

```
1  DBD-ODBC.INFO=Required for ODBC-
2  DBD-Oracle.INFO=Required for Ora
3  DBD-Pg.INFO=Required for Postgre
4  DBD-mysql.INFO=Required for MyS
5  DBNAME.NAME=Database
6  DBTABLE.NAME=Table Name
7  ENABLE_STREAM.NAME=Enable strea
8  ENCODING.NAME=Encoding
9  HELP=org.talend.help.tMysqlInput
10 HOST.NAME=Host
11 LONG_NAME=Reads a MySQL table a
12 NB_LINE.NAME=Number of line
13 NULL_CHAR.NAME=Null Char
14 PASS.NAME=Password
15 PORT.NAME=Port
16 PROPERTIES.NAME=Additional JDBC
17 QUERY.NAME=Query
18 QUERYSTORE.NAME=Query Type
```

tMysqlInput_messages_fr.properties

```
1  DBD-ODBC.INFO=Requis pour les cor
2  DBD-Oracle.INFO=Requis par Oracle
3  DBD-Pg.INFO=Requis par PostgreSQL
4  DBD-mysql.INFO=Requis par MySQL
5  DBNAME.NAME=Base de données
6  DBTABLE.NAME=Nom de table
7  ENABLE_STREAM.NAME=Activer la dif
8  ENCODING.NAME=Encodage
9  HELP=org.talend.help.tMysqlInput
10 HOST.NAME=Hôte
11 LONG_NAME=Lit une table MySQL et
12 NB_LINE.NAME=Nombre de ligne
13 NULL_CHAR.NAME=Caractère Null
14 PASS.NAME=Mot de passe
15 PORT.NAME=Port
16 PROPERTIES.NAME=Paramètres JDBC a
17 QUERY.NAME=Requête
```

Déposer et relier les composants

1. Cliquez-glissez deux composants **tFileInputProperties** de la famille **File** > **Input** dans le Job designer, un composant **tMap** de la famille **Processing** et un composant **tLogRow** de la famille **Log & Errors**.

2. Reliez les composants entre eux via des connexions **Row** > **Main** Le second fichier, *FR*, est utilisé comme flux de référence.

Configurer les composants

1. Double-cliquez sur le premier composant **tFileInputProperties** pour paramétrer ses propriétés dans l'onglet **Basic settings** de la vue **Component**.

2. Dans le champ **File Format**, sélectionnez le format de votre fichier.

3. Dans le champ **File Name**, cliquez sur le bouton **[...]** et sélectionnez le fichier *.properties* anglais que vous souhaitez utiliser.

4. Répétez cette procédure avec le deuxième composant et le fichier français :

5. Double-cliquez sur le composant **tMap** pour ouvrir son éditeur :

6. Sélectionnez toutes les colonnes de la table *English_terms* et glissez-les vers la table de sortie.

   Sélectionnez la colonne *key* de la table *English_terms* et glissez-la dans la colonne *key* de la table *French_terms*.

7. Dans le tableau **glossary** de la zone **Schema editor** en bas du **tMap**, renommez le champ *value* en *EN* puisqu'il contiendra les valeurs du fichier anglais.

8. Cliquez sur le bouton **[+]** pour ajouter une ligne au tableau **glossary** et renommez-la *FR*.

9. Dans le champ **Length**, configurez la taille maximum du champ à *255*.

10. Dans le coin supérieur gauche de l'éditeur du **tMap**, sélectionnez la colonne *value* de la table *English_terms* et glissez-la dans la colonne *FR* de la table *French_terms*. Cliquez sur **OK** pour valider vos modifications et fermer l'éditeur de mapping, puis propagez les modifications au composant suivant.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour enregistrer votre Job.

2. Appuyez sur **F6** pour l'exécuter ou cliquez sur le bouton **Run** de la vue **Run**.

```
Starting job tFileInputProperties at 15:25 29/05/2009.

PORT.NAME | Port | Port
HELP | org.talend.help.tMysqlInput | org.talend.help.tMysqlInput
STRING_QUOTE.NAME | String Quote | Séparateur de chaine de
caractère
DBTABLE.NAME | Table Name | Nom de table
QUERYSTORE.NAME | Query Type | Type de requête
SQL_SYNTAX.NAME | Sql Syntax | Syntaxe SQL
TYPE.ITEM.PGSQL | PostgreSQL | PostgreSQL
TYPE.NAME | Database Driver | Pilote de base de données
TABLE.NAME | Table Name | Nom de table
ENCODING.NAME | Encoding | Encodage
QUERY.NAME | Query | Requête
DBD-ODBC.INFO | Required for ODBC-like connection | Requis pour
les connexions de type ODBC
```

Le glossaire apparaît dans la console de la vue **Run** avec dans la première colonne le nom de la clé, dans la deuxième le terme du fichier anglais et dans le troisième colonne le terme correspondant du fichier français.

tFileInputRaw

Propriétés du tFileInputRaw

Famille de composant	File/Input	
Fonction	Le composant **tFileInputRaw** lit le contenu d'un fichier et l'envoie dans une colonne de sortie .	
Objectif	Ce composant est utilisé pour rassembler les données et les envoyer dans une colonne de sortie, afin qu'un autre composant puisse les traiter.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Filename	Nom et chemin d'accès au fichier d'entrée à traiter. Vous pouvez saisir le chemin manuellement entre guillemets doubles ou parcourir votre système en cliquant sur le bouton **[...]**.
	Mode	**Read the file as a string** : Le contenu du fichier est lu comme (`string`). **Read the file as a bytes array** : Le contenu du fichier est lu comme `bytes array`. **Stream the file** : Dès que le premier caractère est saisi dans le fichier source, il est immédiatement lu.
	Encoding	Si vous utilisez le mode **Read the file as a string**, sélectionnez le type d'encodage dans la liste ou sélectionnez **Custom** pour le définir manuellement.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**. Pour capturer l'exception `FileNotFoundException`, vous devez cocher cette case.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables	**FILENAME_PATH** : chemin du fichier d'entrée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé pour fournir des données d'entrée pour les Jobs nécessitant une seule colonne de données ou nécessitant qu'un fichier complet soit lu comme une seule colonne.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *Scénario : Télécharger des fichiers dans Dropbox*

tFileInputRegex

Propriétés du tFileInputRegex

Famille de composant	File/Input	
Fonction	Puissant composant qui peut remplacer bon nombre des composants File. Il requiert des connaissances avancées en rédaction d'expressions régulières.	
Objectif	Le **tFileInputRegex** lit un fichier ou un flux de données ligne par ligne afin de le diviser en champs à l'aide d'expressions régulières et d'envoyer les champs définis dans le schéma au composant suivant via une connexion Row.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	File name/Stream	**File name** : Chemin d'accès et nom du fichier, et/ou variable à traiter. **Stream** : Flux de données à traiter. Les données doivent préalablement être mises en flux afin d'être récupérées par le **tFileInputRegex** via la variable *INPUT_STREAM* disponible dans la liste d'auto-complétion (**Ctrl+Espace**). Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*. Notez que dans la version Map/Reduce du **tFileInputRegex**, le champ à utiliser pour lire les fichiers est **Folder/File**.
	Folder/File	Parcourez votre système ou saisissez le chemin d'accès au répertoire de HDFS dans lequel sont stockées les données à utiliser. Si le chemin d'accès défini pointe vers un dossier, ce composant lit tous les fichiers stockés dans le dossier, par exemple */user/talend/in*. Si des sous-dossiers existent, ils sont automatiquement ignorés à moins que le chemin défini soit */user/talend/in/**. Si vous souhaitez spécifier plusieurs fichiers ou dossiers dans ce champ, séparez les chemins à l'aide d'une virgule (,). Si le fichier à lire est un fichier compressé, saisissez son nom ainsi que son extension. Le **tHDFSInput** décompresse le fichier automatiquement à l'exécution. Les formats de compression, ainsi que les extensions correspondantes, sont : • DEFLATE : **.deflate* • gzip : **.gz* • bzip2 : **.bz2* • LZO : **.lzo* Assurez-vous que la connexion à la distribution Hadoop utilisée est correctement configurée dans l'onglet **Hadoop configuration** de la vue **Run**.
	Row separator	Saisissez le séparateur à utiliser pour identifier la fin des lignes.

	Regex	Ce champ est compatible peut contenir plusieurs lignes. Intégrez à vos expressions régulières le subpattern correspondant aux champs à extraire. En Java, doublez les antislashes en regexp.
	Header	Saisissez le nombre de lignes à ignorer au début du fichier.
	Footer	Nombre de lignes à ignorer en fin de fichier.
	Limit	Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n'est lue ou traitée.
	Ignore error message for the unmatched record	Cochez cette case afin de ne pas écrire en sortie les messages d'erreur pour les enregistrements qui ne correspondent pas à l'expression régulière définie. Par défaut, cette case n'est pas cochée.
	Schema et *Edit Schema*	Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et stocké dans le Repository. Ainsi, il est réutilisable dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Skip empty rows	Cochez cette case pour ignorer les lignes vides.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**.
Advanced settings	*Encoding*	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données. In the Map/Reduce version of **tFileInputRegex**, you need to select the **Custom encoding** check box to display this list.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. La variable **NB_LINE** n'est pas disponible dans la version Map/Reduce. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant sert à lire un fichier et à en séparer les champs selon les Regex définies. Ce composant permet de créer un flux de données à l'aide d'un lien **Row** > **Main**, ainsi que de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d'utilisation de ces deux liens, consultez *Scénario 2 : Extraire les données XML erronées dans un flux de rejet* du composant **tFileInputXML**.
Utilisation dans des Jobs Map/Reduce	Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme composant de début et requiert un composant de transformation comme lien de sortie. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Transformer en fichier Regex en Positional

Le scénario suivant est un Job formé de deux composants, qui a pour objectif de lire les données d'un fichier d'entrée utilisant des expressions régulières et transformant les données ainsi délimitées en sortie positionnelle.

Déposer et relier les composants

1. Cliquez et déposez un composant **tFileInputRegex** et **tFileOutputPositional** de la **Palette** dans l'espace de modélisation.

2. Cliquez-droit sur le composant **tFileInputRegex** et sélectionnez une connexion **Row** > **Main** dans le menu contextuel. Faites glisser cette connexion vers le composant **tFileOutputPositional** et relâchez lorsque le symbole approprié apparaît.

1. Sélectionnez **tFileInputRegex** à nouveau et définissez ses propriétés dans l'onglet **Component** :

2. Les propriétés de ce Job sont de type built-in. Par conséquent, elles ne sont définies que pour ce Job et ne peuvent être partagées avec d'autres utilisateurs.

3. Indiquez le chemin d'accès au fichier dans le champ **File Name**. Ce champ est obligatoire.

4. Définissez le séparateur de lignes, dans le champ **Row separator**, afin d'identifier la fin d'une ligne.

5. Puis, dans **Regular expression**, définissez l'expression régulière à utiliser pour délimiter les champs d'une ligne qui seront passés au composant suivant. Vous pouvez saisir une expression régulière multiligne si vous en avez besoin.

6. Assurez-vous d'inclure dans cette expression tous les sous-patterns des champs à extraire.

7. Dans ce scénario, ignorez les champs en-tête (**Header**), pied de page (**Footer**) et limite (**Limit**).

8. Dans le **Schema**, sélectionnez local (**Built-in**) comme type de schéma des données à transférer au composant **tFileOutputPositional**.

9. Vous pouvez charger ou créer à partir de la fonction **Edit Schema**.

10. Puis définissez les propriétés du deuxième composant :

11. Saisissez le chemin d'accès au fichier de sortie.

12. Saisissez l'encodage du fichier de sortie dans le champ **Encoding**. Notez que pour l'instant, l'homogénéité de l'encodage dans un Job n'est pas vérifiée.

13. Sélectionnez le type de schéma dans **Schema**. Cliquez sur le bouton **Sync columns** pour récupérer en sortie le schéma d'entrée.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Passez à l'onglet **Run** et cliquez sur le bouton **Run** pour exécuter le Job.

Le fichier d'entrée est lu ligne par ligne puis divisé en champs délimités selon la valeur de l'expression régulière définie. Vous pouvez ouvrir le fichier de sortie avec tout éditeur standard.

tFileInputXML

Le **tFileInputXML** appartient à deux familles de composants : File et XML. Pour plus d'informations concernant le composant **tFileInputXML**, consultez *tFileInputXML*.

tFileList

Propriétés du tFileList

Famille de composant	File/Management	
Fonction	Le composant **tFileList** fait une boucle sur les fichiers et/ou les dossiers d'un répertoire donné.	
Objectif	Le **tFileList** récupère un groupe de fichiers et/ou de dossiers suivant un masque donné et fait une boucle sur chacun d'eux.	
Basic settings	*Directory*	Chemin d'accès au répertoire de fichiers.
	FileList Type	Sélectionnez à partir de la liste déroulante le type de source d'entrée utilisé : **Files** pour des fichiers, **Directories** pour des répertoires ou **Both** pour les deux types.
	Include subdirectories	Cochez cette case si le ou les répertoires sélectionné(s) contiennent des sous-dossiers.
	Case Sensitive	Cochez cette case pour créer un filtre selon la casse des noms des fichiers.
	Generate Error if no file found	Cochez cette case pour être averti par un message d'erreur que la recherche de fichier n'a pas abouti.
	Use Glob Expressions as Filemask (Unchecked means Perl5 Regex Expressions)	Cette case est cochée par défaut et permet de filtrer les résultats à l'aide d'expressions globales (**Glob Expressions**). Décochez la case pour filtrer les résultats à l'aide d'expressions de type regex Perl5.
	Files	Cliquez sur le bouton **[+]** pour ajouter des lignes de filtre : **Filemask** : entrez un nom de fichier ou masque de fichier utilisant un caractère spécial ou une expression régulière.
	Order by	Les dossiers sont listés en premier, puis viennent ensuite les fichiers. Vous pouvez définir les priorités des dossiers et des fichiers de différentes manières : **By default** : par ordre alphabétique, par dossier puis par fichier, **By file name** :par ordre alphabétique ou ordre alphabétique inversé, **By file size** : du plus petit léger au plus lourd ou du plus lourd au plus léger, **By modified date** : du plus récent au plus ancien ou du plus ancien au plus récent. Lorsque vous sélectionnez l'option **By file name**, si deux fichiers ont le même nom, le classement s'effectuera via l'option **By modified date**. Si vous choisissez l'option **By file size**, et que les fichiers font le même poids, ils seront classés selon l'option **By file name**. Si vous sélectionnez l'option **By modified date** et que la date de modification est la même pour deux fichiers, alors ils seront classés selon l'option **By file name**.
	Order action	Choisissez de classer par ordre : **ASC** : ordre alphabétique / du plus petit au plus grand / du plus ancien au plus récent. **DESC** : à l'inverse de l'ordre alphabétique / du plus grand au plus petit / du plus ancien au plus récent.

Advanced settings	*Use Exclude Filemask*	Cochez cette case pour activer le champ **Exclude Filemask** afin d'exclure les conditions de filtre basées sur le type de fichier : **Exclude Filemask** : Renseignez le champ en saisissant les types de fichier devant être exclus du tableau **Files** de la vue **Basic settings**. Les types de fichier dans ce champ doivent être entourés de guillemets doubles et séparés par des virgules.
	Format file path to slash (/) style (useful on Windows)	Cochez cette case afin de formater le chemin d'accès au fichier en insérant des barres obliques (/), option utile sous Windows.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Utilisation	Le composant **tFileList** fournit une liste de fichiers d'un répertoire donné sur lequel il fait une boucle.	
Global Variables	**CURRENT_FILE** : nom du fichier courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_FILEPATH** : nom du fichier courant ainsi que son chemin d'accès. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_FILEEXTENSION** : extension du fichier courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_FILEDIRECTORY** : répertoire du fichier courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_FILE** : nombre de fichiers itérés. Cette variable est une variable **Flow** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Iterate. **Trigger** : OnSubjobOk, OnSubjobError, OnComponentOk, OnComponentError. Les liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate. **Trigger** : Run If, OnSubjobOk, OnSubjobError, Synchronize, Parallelize. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Itération sur un répertoire

Le scénario suivant décrit un Job de trois composants, qui a pour objectif de répertorier les fichiers d'un répertoire donné, de lire chaque fichier par itération, de sélectionner les données délimitées et d'afficher ces données en sortie dans la console **Run**.

Déposer et relier les composants

1. Cliquez et déposez les composants suivants : **tFileList**, **tFileInputDelimited** et **tLogRow** dans l'espace de modélisation.

2. Cliquez-droit sur le composant **tFileList** et connectez-le au composant **tFileInputDelimited** via une connexion de type **Row Iterate**. Puis connectez le **tFileInputDelimited** au **tLogRow** à l'aide d'une connexion de type **Row Main**.

Configurer les composants

1. Sélectionnez le composant **tFileList** et cliquez sur l'onglet **Basic settings** :

2. Dans le champ **Directory**, indiquez le chemin d'accès des fichiers à traiter. Pour faire apparaître ce chemin d'accès dans le Job lui-même, utilisez le nom de champ (__DIRECTORY__) qui apparaît en bulle info lorsque vous passez le souris sur le champ **Directory**. Saisissez cette référence dans le champ **Label Format** de l'onglet **View**.

3. Dans le champ **FileList Type**, sélectionnez l'option **Files**.

4. Définissez le champ **Case sensitive** sur **Yes** pour prendre la casse en compte dans cet exemple.

5. Laissez la case **Use Glob Expressions as Filemask** cochée si vous utilisez des expressions globales pour filtrer les fichiers et définissez un masque de fichier dans le champ **Filemask**.

6. Double-cliquez sur le **tFileInputDelimited** afin d'ouvrir sa vue **Basic settings** et définissez ses propriétés.

7. Dans le champ **File Name**, nommez le fichier à l'aide d'une variable contenant son chemin d'accès, tel que vous l'avez renseigné dans les propriétés du **tFileList**. Appuyez sur **Ctrl +Espace** pour accéder à la liste d'autocomplétion des variables et sélectionnez la variable globale `((String)globalMap.get("tFileList_1_CURRENT_FILEPATH"))`.

8. Renseignez les autres champs comme pour un fichier délimité **tFileInputDelimited** standard. Voir également : *tFileInputDelimited*.

9. Sélectionnez le composant de sortie **tLogRow** et renseignez le séparateur de champs qui s'affichera dans la console de l'onglet **Run**. Voir également : *Propriétés du tLogRow*.

Sauvegarder et exécuter le Job

Appuyez sur **Ctrl+S** pour sauvegarder votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job test at 17:16 21/09/2009.

.--+-------------------------------+-----------------------+-------.
|                              tLogRow_1                              |
|=-+-------------------------------+-----------------------+------=|
|id|CustomerName                   |CustomerAddress        |idState|
|=-+-------------------------------+-----------------------+------=|
|1 |Griffith Paving and Sealcoatin|talend@apres91         |7      |
|2 |Bill's Dive Shop               |511 Maple Ave.  Apt. 1B|35     |
|3 |Childress Child Day Care       |662 Lyons Circle       |1      |
|4 |Facelift Kitchen and Bath      |220 Vine Ave.          |41     |
|5 |Terrinni & Son Auto and Truck |770 Exmoor Rd.          |5      |
|6 |Kermit the Pet Shop            |1860 Parkside Ln.      |28     |
|7 |Tub's Furniture Store          |807 Old Trail Rd.      |15     |
|8 |Toggle & Myerson Ltd           |618 Sheriden rd.       |9      |
|9 |Childress Child Day Care       |788 Tennyson Ave.      |12     |
|10|Elle Hypnosis and Therapy Cent|2032 Northbrook Ct.     |1      |
'--+-------------------------------+-----------------------+-------'
.--+-------------------------------+-----------------------+-------.
|                              tLogRow_1                              |
|=-+-------------------------------+-----------------------+------=|
|id|CustomerName                   |CustomerAddress        |idState|
|=-+-------------------------------+-----------------------+------=|
|1 |Glenwood Credit Union          |511 Maple Ave.  Apt. 1B|46     |
|2 |Gourmet the Frog               |788 Tennyson Ave.      |1      |
|3 |Acturial Enterprises Ltd.      |3385 University Ave.   |34     |
|4 |Salt & Pepper Catering Service|965 Marion Place Apt. 65C|44    |
|5 |Rythmics Ltd.                  |1875 Roger Williams Ave.|22    |
|6 |Parkway Auto Body              |1859 Green Bay Rd.     |27     |
|7 |Futoons Cartoons Emporium      |1486 Old Deerfield Rd. |24     |
|8 |Glenn Oaks Office Supplies     |1882 St. Johns         |10     |
|9 |Garfield Appliance Service     |3150 Skokie Valley Rd. |33     |
|10|New Dehli Auto Exchange        |1957 Huntington Ave.   |49     |
'--+-------------------------------+-----------------------+-------'
Job test ended at 17:16 21/09/2009. [exit code=0]
```

L'itération s'effectue sur le répertoire défini. Chacun des fichiers contenus dans ce répertoire est lu. Puis les données délimitées sont transmises au composant de sortie qui les affiche dans la console Log de la vue **Run**.

Scénario 2 : Trouver des fichiers dupliqués entre deux dossiers

Le scénario suivant décrit un Job qui fait une boucle sur deux dossiers, transforme les résultats de l'itération en flux de données afin d'obtenir une liste de noms de fichiers puis affiche les doublons dans la console **Run**. Ce scénario peut servir de préparation pour la fusion de deux dossiers par exemple.

Déposer et relier les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : deux **tFileList**, deux **tIterateToFlow**, deux **tFileOutputDelimited**, un **tFileInputDelimited**, un **tUniqRow** et un **tLogRow**.

2. Reliez le premier **tFileList** au premier **tIterateToFlow** à l'aide d'un lien **Row** > **Iterate** et reliez le premier **tIterateToFlow** au premier **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main** afin de former le premier sous-Job.

3. Reliez le second **tFileList** au second **tIterateToFlow** à l'aide d'un lien **Row** > **Iterate** et reliez le second **tIterateToFlow** au second **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main** afin de former le deuxième sous-Job.

4. Reliez le **tFileInputDelimited** au **tUniqRow** à l'aide d'un lien **Row** > **Main** et reliez le **tUniqRow** au **tLogRow** à l'aide d'un lien **Row** > **Duplicates** afin de former le troisième sous-Job.

5. Reliez les trois sous-Jobs à l'aide de liens **Trigger** > **On Subjob Ok** afin qu'ils soient déclenchés l'un après l'autre. Nommez ensuite les composants afin de mieux identifier leur rôle dans le Job.

Configurer les composants

1. Dans le champ **Directory** de la vue **Basic settings** du premier **tFileList**, saisissez le chemin vers le premier dossier dans lequel vous souhaitez lire le nom des fichiers, *E:/DataFiles/DI/images* dans ce scénario. Laissez les autres paramètres tels qu'ils sont.

2. Double-cliquez sur le premier **tIterateToFlow** afin d'afficher sa vue **Basic settings**.

3. Double-cliquez sur le bouton **[...]** situé à côté de **Edit schema** afin d'ouvrir la boîte de dialogue **[Schema]**. Définissez le schéma du fichier texte dans lequel le second composant doit écrire les noms des fichiers. Une fois fait, cliquez sur **OK** afin de fermer la boîte de dialogue et propager les changements au composant suivant.

   Dans cet exemple, le schéma contient une seule colonne : *Filename*.

4. Dans le champ **Value** de la table **Mapping**, appuyez sur **Ctrl+Espace** afin d'ouvrir la liste d'autocomplétion des variables. Sélectionnez la variable globale `((String)globalMap.get("tFileList_1_CURRENT_FILE"))` afin de lire le nom de chaque fichier du dossier d'entrée. Ce dernier sera ensuite inséré dans un flux de données et passé au composant suivant.

5. Dans le champ **File Name** de la vue **Basic settings** du premier **tFileOutputDelimited**, saisissez le chemin vers le fichier texte contenant les noms des fichiers provenant du flux entrant, *D:/temp/tempdata.csv* dans cet exemple. La configuration du premier sous-Job est ainsi terminée.

6. Configurez le deuxième sous-Job de la même manière que le premier sous-Job. Cependant :

   - Dans le champ **Directory** de la vue **Basic settings** du second **tFileList**, saisissez le chemin vers l'autre dossier dans lequel vous souhaitez lire les noms des fichiers, *E:/DataFiles/DQ/images* dans ce scénario.

   - Dans la vue **Basic settings** du second **tFileOutputDelimited**, cochez la case **Append** afin que les noms des fichiers précédemment écrits dans le fichier texte ne soit pas écrasés.

7. Dans le champ **File name/Stream** de la vue **Basic settings** du **tFileInputDelimited**, saisissez le chemin vers le fichier texte contenant la liste de noms de fichiers, *D:/temp/tempdata.csv* dans ce scénario. Définissez le schéma du fichier qui, dans le présent exemple, contient une seule colonne, *Filename*.

8. Dans la vue **Basic settings** du **tUniqRow**, cochez la case **Key attribute** pour la colonne, *Filename* in this exmaple.

9. Dans la vue **Basic settings** du **tLogRow**, sélectionnez l'option **Table (print values in cells of a table)** pour un meilleur affichage des résultats d'exécution du Job.

Exécuter le Job

1. Appuyez sur **Ctrl+S** afin de sauvegarder le Job.

2. Cliquez sur **Run** ou appuyez sur **F6** afin d'exécuter le Job.

   Tous les fichiers dupliqués entre les deux dossiers sélectionnés sont affichés dans la console.

```
[statistics] connected
.-------------------------.
|    Duplicates DI-DQ     |
|=------------------------=|
|Filename                 |
|=------------------------=|
|chapter_header1.jpg      |
|db_connection_teradata.png|
|Delete_Button.png        |
|di_dq_mdm_ai_cover.jpg   |
|draft.png                |
|Import_Items1.png        |
|Import_Items2.png        |
|mdm_connection.png       |
|mdm_connection1.png      |
|mdm_connection2.png      |
|mdm_connection3.png      |
|mdm_connection4.png      |
|mdm_connection5.png      |
|note.png                 |
|svn_user_log.png         |
|warning.png              |
'-------------------------'
[statistics] disconnected
```

Pour d'autres scénarios utilisant le **tFileList**, *tFileCopy*.

tFileOutputARFF

Propriétés du tFileOutputARFF

Famille de composant	File/Output	
Fonction	Le composant **tFileOutputARFF** transmet des données vers un fichier ARFF.	
Objectif	Ce composant écrit un fichier ARFF contenant des données organisées en fonction du schéma défini.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant connexion de base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant. Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le *Guide utilisateur du Studio Talend*.
	File name	Chemin d'accès et nom du fichier de sortie, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Attribute Define	Reflète le schéma tel qu'il est défini dans la boîte de dialogue **[Edit schema]**. **Column** : Nom de la colonne. **Type** : Type des données de la colonne. **Pattern** : Saisissez le modèle (pattern) de la donnée si nécessaire.
	Relation	Saisissez le nom de la relation.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider

		de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
Advanced settings	*Don't generate empty file*	Cochez cette case pour empêcher la génération d'un fichier vide.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : On Subjob Ok; On Subjob Error; Run if. Liens d'entrée (d'un autre composant à celui-ci) : **Row**: Main; Reject; Iterate. **Trigger**: On Subjob Ok; On Subjob Error; Run if; On Component Ok; On Component Error; Synchronize; Parallelize. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour écrire dans un fichier ARFF des données récupérées d'autres composants à l'aide d'une connexion de type **Row**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *Scénario : Afficher le contenu d'un fichier ARFF*.

tFileOutputDelimited

Propriétés du tFileOutputDelimited

Famille de composant	File/Output	
Fonction	Le composant **tFileOutputDelimited** transmet des données vers un fichier délimité. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job Map/Reduce **Talend** afin de générer du code Map/Reduce. Pour plus d'informations, consultez *Le tFileOutputDelimited dans des Jobs Map/Reduce Talend*.	
Objectif	Ce composant écrit un fichier délimité contenant des données organisées en fonction du schéma défini.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use Output Stream	Cochez la case pour traiter le flux de données qui vous intéresse. Une fois cochée, le champ **Output Stream** s'affiche et vous pouvez saisir le flux de données souhaité. Le flux de données à traiter doit être ajouté au flux afin que ce composant récupère ces données via la variable représentative correspondante. Cette variable peut être prédéfinie dans votre Studio ou fournie par le contexte ou les composants utilisé(s) avec ce composant. Sinon, vous pouvez la définir manuellement et l'utiliser selon votre Job, par exemple à l'aide d'un **tJava** ou d'un **tJavaFlex**. Afin d'éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d'autocomplétion (**Ctrl+Espace**) afin de remplir le champ, si cette variable a été correctement définie. Pour plus d'informations concernant l'utilisation d'un flux, consultez *Scénario 2 : Lire les données d'un fichier distant en mode stream*.
	File name	Chemin d'accès et nom du fichier de sortie et/ou variable à utiliser. Ce champ est indisponible lorsque vous avez coché la case **Use Output Stream**. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Row Separator	Saisissez le séparateur à utiliser pour identifier la fin des lignes.
	Field Separator	Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.
	Append	Cochez cette option pour ajouter de nouvelles lignes à la fin du fichier.
	Include Header	Cochez cette case pour tenir compte de l'en-tête dans le fichier de sortie.
	Compress as zip file	Cochez cette case pour compresser en zip le fichier de sortie.

	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Cliquez sur ce bouton pour synchroniser le schéma de sortie avec celui d'entrée. La fonction **Sync** ne s'affiche que si une connexion de type **Row** est liée au composant de sortie.
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	CSV options	Cochez cette case pour inclure des paramètres CSV spécifiques comme **Escape char** et **Text enclosure**.
	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Split output in several files	Si le fichier délimité en sortie est volumineux, vous pouvez le scinder en plusieurs fichiers. **Rows in each output file** : saisissez le nombre de lignes pour chaque fichier de sortie.
	Custom the flush buffer size	Cochez cette case pour personnaliser le nombre de lignes à écrire avant de vider le tampon. **Row Number** : Nombre de lignes à écrire avant de vider le tampon.
	Output in row mode	Cochez cette case pour que l'écriture se fasse en mode ligne.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
	Don't generate empty file	Cochez cette case pour empêcher la génération d'un fichier vide.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **FILE_NAME** : nom du fichier traité. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour écrire dans un fichier délimité des données récupérées d'autres composants à l'aide d'une connexion de type **Row** et séparer les champs à l'aide d'un séparateur.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Le tFileOutputDelimited dans des Jobs Map/Reduce Talend

*Les informations contenues dans cette section concernent uniquement les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend** et ne sont pas applicables aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job Map/Reduce **Talend**, **tFileOutputDelimited**, ainsi que le Job complet utilisant ce composant, génère du code Map/Reduce natif. Cette section présente les propriétés spécifiques du **tFileOutputDelimited** lorsqu'il est utilisé dans un Job Map/Reduce. Pour plus d'informations concernant un Job Map/Reduce **Talend**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	MapReduce/Output	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Réutilisez les propriétés stockées centralement sous le nœud **Hadoop Cluster** de la vue **Repository**. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Pour plus d'informations concernant le nœud **Hadoop Cluster**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.
		Cliquez sur cette icône pour ouvrir l'assistant de connexion à la base de données et stocker les paramètres de connexion configurés dans la vue **Basic settings**. Pour plus d'informations concernant la configuration et le stockage des paramètres de connexion à la base de données, consultez le *Guide utilisateur du* Studio Talend.

	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Folder	Parcourez votre système ou saisissez le chemin d'accès au répertoire de HDFS dans lequel sont stockées les données à utiliser. Ce chemin d'accès doit pointer vers un dossier et non un fichier, car un Job Map/Reduce **Talend** doit écrire dans un dossier cible le résultat final mais aussi les différents fichiers *part-* générés durant les calculs Map/Reduce. Assurez-vous que la connexion à la distribution Hadoop utilisée est correctement configurée dans l'onglet **Hadoop configuration** de la vue **Run**.
	Action	Sélectionnez une opération pour écrire les données : **Create** : crée un fichier et écrit des données dans ce fichier. **Overwrite** : écrase le fichier existant dans le répertoire spécifié dans le champ **Folder**.
	Row separator	Saisissez le séparateur à utiliser pour identifier la fin des lignes.
	Field separator	Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.
	Include Header	Cochez cette case pour inclure l'en-tête de la colonne dans le fichier.
	Custom encoding	Il est possible de rencontrer des problèmes d'encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste **Encoding**. Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
	Compress the data	Cochez la case **Compress the data** afin de compresser les données de sortie. Hadoop fournit différents formats de compression permettant de réduire l'espace nécessaire au stockage des fichiers et d'accélérer le transfert de données. Lorsque vous lisez un fichier compressé, le *Studio Talend* doit le décompresser avant de pouvoir en alimente le flux d'entrée.
	Merge result to single file	Cochez cette case pour fusionner les fichiers finaux *part* en un fichier et mettre ce fichier dans un répertoire spécifié.

		Une fois cochée, saisissez le chemin d'accès ou parcourez votre système jusqu'au dossier dans lequel stocker le fichier fusionné. Ce répertoire est automatiquement créé s'il n'existe pas. Les cases suivantes sont utilisées pour gérer les fichiers source et cible : • **Remove source dir** : cochez cette case pour supprimer les fichiers source après la fusion. • **Override target file** : cochez cette case pour écraser le fichier existant à l'emplacement cible. Cette option n'écrase pas le dossier. Ce champ n'est pas disponible pour un fichier Sequence.
Advanced settings	*Advanced separator (for number)*	Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.). Ce champ n'est pas disponible pour un fichier Sequence.
	CSV options	Cochez cette case pour inclure des paramètres CSV spécifiques comme **Escape char** et **Text enclosure**.
	Enable parallel execution	Cocher cette case permet de traiter plusieurs flux de données simultanément. Cela permet de traiter les données plus rapidement. Notez que cette fonctionnalité dépend de la capacité de la base de données ou de l'application à gérer plusieurs insertions en parallèle ainsi que le nombre de processeurs utilisés. Dans le champ **Number of parallel executions**, vous pouvez : • Saisir le nombre d'exécutions parallèles désiré. • Appuyer sur **Ctrl+Espace** et sélectionner la variable de contexte dans la liste. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Dans un Job **Talend** Map/Reduce, ce composant est utilisé en tant que composant de fin et requiert un composant de transformation comme lien d'entrée. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Lorsqu'un Job Map/Reduce est ouvert dans l'espace de modélisation graphique, le **tFileOutputDelimited** et la famille **MapReduce** s'affichent dans la **Palette** du Studio. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.	

Scénario 1 : Ecrire des données dans un fichier délimité

Ce scénario est constitué de trois composants et a pour objectif d'extraire certaines données d'un fichier contenant des informations sur les clients (*customers*) puis de les écrire dans un fichier délimité en sortie.

Dans l'exemple suivant, vous avez enregistré les informations de connexion et les métadonnées dans le **Repository** afin de faciliter leur réutilisation. Pour plus d'informations sur comment centraliser les données de connexion et autres métadonnées d'un fichier dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

Déposer et relier les composants

1. Dans le **Repository**, ouvrez le nœud **File Delimited** de la catégorie **Metadata**. Cliquez sur la métadonnée *customers*, et déposez-la dans l'espace de modélisation graphique. Une boîte de dialogue s'ouvre et vous demande quel type de composant vous souhaitez déposer.

2. Cliquez sur **tFileInputDelimited** puis sur **OK**. Un composant **tFileInputDelimited** contenant le nom de votre schéma d'entrée s'affiche dans l'espace de modélisation graphique.

3. Déposez un composant **tMap** et un composant **tFileOutputDelimited** de la **Palette** dans l'espace de modélisation graphique.

4. Reliez les composants à l'aide d'un lien **Row** > **Main**.

Configurer les composants

Configurer le composant d'entrée

1. Double-cliquez sur le composant **tFileInputDelimited** afin d'afficher sa vue **Basic settings**. Tous les champs sont renseignés automatiquement car vous avez défini votre fichier d'entrée localement.

2. Si vous ne définissez pas votre fichier d'entrée localement dans la vue **Repository**, renseignez les détails manuellement après avoir sélectionné **Built-in** dans la liste **Property Type**.

3. Cliquez sur le bouton **[...]** près du champ **File Name** et dans la boîte de dialogue de sélection, parcourez le système de fichiers et sélectionnez le fichier d'entrée, dans cet exemple: *customer.csv*.

   Si le chemin d'accès du fichier contient des caractères accentués, vous obtiendrez un message d'erreur lors de l'exécution du Job. Pour plus d'informations concernant la procédure à suivre lorsque les accents ne sont pas supportés, consultez le Guide d'installation et de migration Talend.

4. Dans les champs **Row Separator** et **Field Separator**, saisissez respectivement le séparateur de lignes "*\n*" et le séparateur de champs "*;*".

5. Si besoin, définissez dans les champs correspondants le nombre de lignes utilisées dans l'en-tête (**Header**) et dans le pied-de-page (**Footer**), puis définissez le nombre de lignes traitées.

   Dans cet exemple, saisissez *6* pour **Header** et laissez par défaut les champs **Footer** et **Limit**.

6. Dans le champ **Schema**, le schéma est défini automatiquement sur **Repository** et votre schéma est déjà défini puisque vous avez stocké votre fichier d'entrée localement pour cet exemple. Sinon, sélectionnez **Built-in** et cliquez sur le bouton **[...]** près de **Edit Schema** pour ouvrir la boîte de dialogue **[Schema]**, dans laquelle vous pouvez définir le schéma d'entrée. Cliquez ensuite sur **OK** pour fermer la boîte de dialogue.

Configurer le composant de mapping

1. Double-cliquez sur le **tMap** afin d'ouvrir son éditeur.

2. Dans cet éditeur, cliquez sur le bouton en haut à droite pour ouvrir la boîte de dialogue **[Add a new output table]**.

3. Saisissez un nom pour la table que vous souhaitez créer, *row2* dans cet exemple.

4. Cliquez sur **OK** pour valider vos modifications et fermer la boîte de dialogue.

5. Dans la table de gauche (*row1*), sélectionnez les trois premières lignes (*Id*, *CustomerName* et *CustomerAddress*) en appuyant sur **Ctrl**, puis glissez-les dans la table de droite.

6. Dans le **Schema editor** en bas de l'éditeur, changez le **Type** de la ligne **RegisterTime**, dans la table de droite, et passez-le en chaîne de caractères, **String**.

Column	Key	Type	N..	Date Pattern ...	Length	Precision	Def...	Comm...
idState		int			2			
id2		int			2			
RegTime		String			30			
RegisterTime		String			30			
Sum1		float			10	5		
Sum2		float			10	5		

7. Cliquez sur **OK** pour enregistrer les modifications et fermer l'éditeur.

Configurer le composant de sortie

1. Dans l'espace de modélisation, double-cliquez sur le composant**tFileOutputDelimited** pour ouvrir sa vue **Component** et définir ses propriétés.

2. Dans le champ **Property Type**, sélectionnez le mode **Built-in** et renseignez les champs suivants manuellement.

3. Cliquez sur le bouton **[...]** correspondant au champ **File Name**. Dans la boîte de dialogue, indiquez le répertoire de destination et le nom du fichier dans lequel vous voulez écrire des données, ici *customerselection.txt*.

4. Pour les champs **Row Separator** et **Field Separator**, saisissez respectivement le séparateur de lignes "*n*" et le séparateur de champs ";". Ils sont utilisés dans le fichier de sortie pour délimiter les valeurs et les lignes.

5. Cochez la case **Include Header** si vous souhaitez récupérer le libellé des colonnes.

6. Cliquez sur **Edit schema** et assurez-vous que le schéma est correctement récupéré du schéma d'entrée. Si nécessaire, cliquez sur **Sync Columns** pour récupérer le schéma du composant précédent.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur **Run** dans l'onglet **Run** pour exécuter le Job.

customerselection.txt

```
1  id;CustomerName;CustomerAddress
2  1;Griffith Paving and Sealcoatin;talend@apres91
3  2;Bill's Dive Shop;511 Maple Ave.  Apt. 1B
4  3;Childress Child Day Care;662 Lyons Circle
5  4;Facelift Kitchen and Bath;unknown
6  5;Terrinni & Son Auto and Truck;770 Exmoor Rd.
7  6;Kermit the Pet Shop;1860 Parkside Ln.
8  7;Tub's Furniture Store;807 Old Trail Rd.
9  8;Toggle & Myerson Ltd;618 Sheriden rd.
10 9;Childress Child Day Care;788 Tennyson Ave.
11 10;Elle Hypnosis and Therapy Cent;2032 Northbrook Ct.
```

Le fichier créé est délimité par des points-virgules, il ne contient que les colonnes *Id*, *CustomerName* et *CustomerAddress*.

Les colonnes *Id*, *CustomerName* et *CustomerAddress* sont écrites dans le fichier de sortie défini.

Scénario 2 : Utiliser un flux de sortie pour sauvegarder des données dans un fichier local

Basé sur le scénario précédent, ce scénario sauvegarde les données filtrées dans un fichier local, à l'aide de l'option **Output Stream**.

Déposer et relier les composants

1. Déposez un composant **tJava** de la **Palette** dans l'espace de modélisation graphique.
2. Connectez le **tJava** au **tFileInputDelimited** à l'aide d'un lien **Trigger** > **On Subjob Ok**.

Configurer les composants

1. Double-cliquez sur le **tJava** pour ouvrir sa vue **Basic settings** et configurer ses propriétés.

2. Dans la zone **Code**, saisissez la commande suivante :

```
new java.io.File("C:/myFolder").mkdirs();
globalMap.put("out_file",new
java.io.FileOutputStream("C:/myFolder/customerselection.txt",false));
```

Dans ce scénario, la commande utilisée dans la zone **Code** du **tJava** va créer un nouveau dossier *C:/myFolder* dans lequel le fichier de sortie *customerselection.txt* sera sauvegardé. Vous pouvez personnaliser la commande selon vos besoins.

3. Double-cliquez sur le **tFileOutputDelimited** afin d'ouvrir sa vue **Basic settings** et configurer ses propriétés.

4. Cochez la case **Use Output Stream** pour activer le champ **Output Stream**, dans lequel vous pouvez définir le flux de sortie, à l'aide d'une commande.

   Renseignez le champ **Output Stream** en saisissant la commande suivante :

```
(java.io.OutputStream)globalMap.get("out_file")
```

Vous pouvez personnaliser la commande dans le champ **Output Stream** en appuyant sur les touches **Ctrl+Espace** pour utiliser l'auto-complétion ou saisissez votre commande manuellement dans le champ. Dans ce scénario, la commande utilisée dans le champ **Output Stream** appelle la classe `java.io.OutputStream` pour écrire le flux de données filtrées dans un fichier local, défini dans la zone **Code** du composant **tJava**.

5. Cliquez sur **Sync columns** afin de récupérer le schéma du composant précédent.

6. Laissez les autres composants tels qu'ils sont.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

   Les trois colonnes spécifiées, *Id*, *CustomerName* et *CustomerAddress* sont écrites dans le fichier de sortie défini.

customerselection.txt

```
id;CustomerName;CustomerAddress
1;Griffith Paving and Sealcoatin;talend@apres91
2;Bill's Dive Shop;511 Maple Ave.  Apt. 1B
3;Childress Child Day Care;662 Lyons Circle
4;Facelift Kitchen and Bath;unknown
5;Terrinni & Son Auto and Truck;770 Exmoor Rd.
6;Kermit the Pet Shop;1860 Parkside Ln.
7;Tub's Furniture Store;807 Old Trail Rd.
8;Toggle & Myerson Ltd;618 Sheriden rd.
9;Childress Child Day Care;788 Tennyson Ave.
10;Elle Hypnosis and Therapy Cent;2032 Northbrook Ct.
```

tFileOutputEBCDIC

Ce composant requiert l'utilisation d'un JDK Oracle.

Propriétés du tFileOutputEBCDIC

Famille de composant	File/Output	Notez que ce composant est déprécié. Les capacités de traitement EBCDIC sont désormais entièrement supportées par *Talend Data Mapper*, qui est inclus avec n'importe quelle solution de **Talend** Platform. Pour plus d'informations concernant *Talend Data Mapper*, consultez le *Guide utilisateur de Talend Data Mapper*.
Fonction	Le composant **tFileOutputEBCDIC** écrit un fichier EBCDIC à partir de plusieurs fichiers de données différents ayant chacun un schéma différent.	
Objectif	Ce composant écrit un fichier EBCDIC avec des données extraites de fichier(s) en fonction des schémas décrivant leur structure.	
Basic settings	*Property type*	Peut être Built-in ou Repository.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema	Sélectionnez une option dans la liste afin d'éditer le schéma des données à traiter. Cette liste est activée lorsque la case **Custom set Original Length in Schema** est cochée.
		Built-in : Sélectionnez cette option pour éditer le schéma Built-in des données à traiter.
		Repository : Sélectionnez cette option pour éditer le schéma du **Repository** sélectionné. Les champs suivants sont automatiquement renseignés à l'aide du schéma sélectionné.
	File Name	Cliquez sur le bouton **[...]** afin de parcourir votre système, ou saisissez le chemin d'accès au fichier EBCDIC contenant les données à générer.
	Edit schema	Cliquez sur le bouton **[...]** pour éditer le schéma Built-in ou Repository des données à traiter.
	Xc2j file	Cliquez sur le bouton **[...]** afin de parcourir votre système, ou saisissez le chemin d'accès au fichier xc2j pour transformer le(s) schéma(s) EBCDIC en un fichier XML intermédiaire. Ce champ est désactivé et le fichier xc2j n'est pas requis lorsque vous cochez la case **Custom set Original Length in Schema**.
	Custom set Original Length in Schema	Cochez cette case pour augmenter la vitesse de lecture des fichiers. Lorsque vous cochez cette case, le champ **Xc2j file** est désactivé et le fichier xc2j n'est pas nécessaire. Vous pouvez éditer le schéma Built-in ou Repository pour les données à traiter.
Advanced settings	*Create directory if not exist*	Cochez cette case pour créer un répertoire lorsque celui spécifié n'existe pas.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour écrire un fichier EBCDIC et séparer les données en fonction des schémas identifiés dans les fichiers en entrée.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Créer un fichier EBCDIC à partir de deux fichiers délimités

Ce scénario utilise l'assistant **[Copybook Connection]** guidant l'utilisateur à travers les différentes étapes de création de connexion au fichier Copybook et de récupération des schémas EBCDIC. L'assistant **[Copybook Connection]** est disponible uniquement pour les utilisateurs ayant souscrit à l'une des solutions *Talend*. Sinon, déposez de la Palette le composant **tFileInputEBCDIC** et configurez ses propriétés manuellement..

Le scénario suivant est un Job de trois composants ayant pour objectif d'écrire un fichier au format EBCDIC à partir de deux fichiers délimités ayant des schémas différents.

Ce scénario Java utilise l'assistant **[EBCDIC Connection]** pour créer une connexion au fichier Copybook et générer un fichier xc2j permettant de récupérer et de transformer les différents schémas du fichier.

- Créez une connexion au fichier Copybook décrivant la structure d'un fichier EBCDIC. Dans ce scénario la connexion Copybook est appelée *EBCDIC*.
- Récupérez le schéma du fichier.

Une fois la connexion au fichier Copybook créée et les schémas récupérés à l'aide des assistants **[EBCDIC]** et **[Schema]**, ils apparaissent sous le nœud **Metadata** > **Copybook**. Ces nouveaux schémas s'appellent *Schema01*, *Schema04* et *Schema05*.

Pour créer un fichier EBCDIC à partir de deux fichiers délimités dans la perspective **Integration** de votre *Studio Talend* :

- Cliquez et déposez deux composants **tFileInputDelimited** et un composant **tFileOutputEBCDIC** dans l'éditeur graphique.
- Pour les relier, cliquez-droit sur chaque composant **tFileInputDelimited**, sélectionnez **Row** > **Main** dans le menu contextuel et cliquez sur le composant **tFileOutputEBCDIC**.
- Double-cliquez sur le premier **tFileInputDelimited** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés :

- Dans le champ **File Name**, sélectionnez le fichier délimité en cliquant sur le bouton **[...]**.

Si le chemin d'accès du fichier contient des caractères accentués, vous obtiendrez un message d'erreur lors de l'exécution du Job. Pour plus d'informations concernant la procédure à suivre lorsque les accents ne sont pas supportés, consultez le Guide d'installation et de migration Talend.

- Dans le champ **Schema**, sélectionnez l'option **Repository**, puis cliquez sur le bouton **[...]** et sélectionnez le schéma correspondant à votre fichier dans la boîte de dialogue, sous le nœud **Copybook**.
- Dans le champ **Header**, indiquez le nombre de ligne en-tête à récupérer et passer au composant suivant : *1* dans ce scénario.
- Paramétrez les propriétés du deuxième **tFileInputDelimited** de la même manière que le premier.
- Double-cliquez sur le **tFileOutputEBCDIC** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés :

- Dans le champ **Data file**, indiquez le répertoire et le nom du fichier EBCDIC qui sera créé à partir des deux fichiers délimités.
- Dans le champ **Xc2j file**, renseignez le chemin d'accès au fichier permettant d'extraire le schéma décrivant la structure du fichier EBCDIC.
- Enregistrez votre Job via **Ctrl+S** et cliquez sur la vue **Run**, cochez les cases **Statistics** et **Exec time** puis cliquez sur **Run** pour exécuter le Job

tFileOutputExcel

Propriétés du tFileOutputExcel

Famille de composant	File/Output	
Fonction	Le composant **tFileOutputExcel** transmet des données vers un fichier de type MS Excel.	
Objectif	Ce composant écrit un fichier MS Excel contenant des données organisées en fonction du schéma défini.	
Basic settings	*Write excel 2007 file format (xlsx)*	Cochez cette case pour écrire les données traitées au format Excel 2007, .xlsx.
	Use Output Stream	Cochez la case pour traiter le flux de données qui vous intéresse. Une fois cochée, le champ **Output Stream** s'affiche et vous pouvez saisir le flux de données souhaité. Le flux de données à traiter doit être ajouté au flux afin que ce composant récupère ces données via la variable représentative correspondante. Cette variable peut être prédéfinie dans votre Studio ou fournie par le contexte ou les composants utilisé(s) avec ce composant. Sinon, vous pouvez la définir manuellement et l'utiliser selon votre Job, par exemple à l'aide d'un **tJava** ou d'un **tJavaFlex**. Afin d'éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d'autocomplétion (**Ctrl+Espace**) afin de remplir le champ, si cette variable a été correctement définie. Pour plus d'informations concernant l'utilisation d'un flux, consultez *Scénario 2 : Lire les données d'un fichier distant en mode stream*.
	File name	Chemin d'accès et nom du fichier de sortie. Ce champ est indisponible lorsque vous avez coché la case **Use Output Stream**. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Sheet name	Nom de la feuille.
	Include header	Cochez cette case pour tenir compte de l'en-tête dans le fichier de sortie.
	Append existing file	Cochez cette option pour ajouter de nouvelles lignes à la fin du fichier. **Append existing sheet** : Cochez cette option pour ajouter de nouvelles lignes à la fin de la feuille Excel.
	Is absolute Y pos.	Cochez cette case pour rajouter des informations dans des cellules préexistantes : **First cell X** : emplacement de la cellule sur l'axe des abscisses. **First cell Y** : emplacement de la cellule sur l'axe des ordonnées. **Keep existing cell format** : lorsque la case est cochée, cette option permet de garder le format et l'alignement de la cellule.
	Font	Sélectionnez dans la liste déroulante la police à utiliser.

	Define all columns auto size	Cette case est décochée par défaut, vous pouvez alors préciser dans le tableau associé quelles sont les colonnes à définir automatiquement. Cochez cette case pour que la taille de toutes les colonnes soit définies automatiquement.
	Schema et Edit Schema	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Cliquez sur ce bouton pour synchroniser le schéma de sortie avec celui d'entrée. La fonction **Sync** ne s'affiche que si une connexion de type **Row** est liée au composant de sortie.
Advanced settings	*Create directory if not exists*	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Custom the flush buffer size	Disponible lorsque la case **Write excel2007 file format** est cochée, dans la vue **Basic settings**. Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ **Row number**, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée.
	Advanced separator (for numbers)	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Don't generate empty file	Cochez cette case pour ne pas générer de fichier vide.
	Recalculate formula	Cochez cette case si vous avez besoin de recalculer une ou des formule(s) dans le fichier Excel spécifié. Cette case apparaît uniquement lorsque vous cochez ces trois cases : **Write excel2007 file format(xlsx)**, **Append existing file** et **Append existing sheet**.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour écrire dans un fichier MS Excel des données récupérées d'autres composants à l'aide d'une connexion de type **Row**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *tSugarCRMInput*.

Pour un scénario relatif à l'utilisation de l'option **Use Output Stream**, consultez *Scénario 2 : Utiliser un flux de sortie pour sauvegarder des données dans un fichier local*.

tFileOutputJSON

Propriétés du tFileOutputJSON

Famille de composant	File /Output	
Fonction	Le composant **tFileOutputJSON** écrit des données dans un fichier structuré JSON de sortie. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job Map/Reduce **Talend** afin de générer du code Map/Reduce. Pour plus d'informations, consultez *Le tFileOutputJSON dans des Jobs Map/Reduce Talend*. Dans ce cas, le **tFileOutputJSON** appartient à la famille de composant MapReduce.	
Objectif	Ce composant reçoit des données et les réécrit dans un bloc de données structurées JSON, dans un ficher de sortie.	
Basic settings	*File Name*	Nom du fichier de sortie et son chemin d'accès.
	Generate an array json	Cochez cette case pour générer un fichier de tableau JSON.
	Name of data blocks	Saisissez, entre guillemets doubles, un nom pour le bloc de données à écrire. Ce champ disparaît lorsque la case **Generate an array json** est cochée.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Cliquez pour synchroniser le schéma du fichier de sortie et le schéma du fichier d'entrée. La fonction **Sync** ne s'affiche que lorsqu'un lien **Row** connecte le composant de sortie.
Advanced settings	*Create directory if not exists*	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.

Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour réécrire des données reçues dans un fichier de sortie structuré JSON.
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Le tFileOutputJSON dans des Jobs Map/Reduce Talend

*Les informations contenues dans cette section concernent uniquement les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend** et ne sont pas applicables aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job Map/Reduce **Talend**, le **tFileOutputJSON**, ainsi que le Job complet utilisant ce composant, génère du code Map/Reduce natif. Cette section présente les propriétés du **tFileOutputJSON** lorsqu'il est utilisé dans un Job Map/Reduce. Pour plus d'informations concernant un Job Map/Reduce **Talend**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	MapReduce / Output	
Fonction	Dans un Job Map/Reduce, le **tFileOutputJSON** reçoit des données d'un composant de transformation et écrit les données dans un ou plusieurs fichiers JSON dans HDFS.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Folder	Saisissez le nom du dossier de HDFS dans lequel vous souhaitez stocker les données du (des) fichier(s) de sortie JSON. Le dossier est automatiquement créé s'il n'existe pas. Assurez-vous que la connexion à la distribution Hadoop utilisée est correctement configurée dans l'onglet **Hadoop configuration** de la vue **Run**.
	Output type	Sélectionnez la structure du (des) fichier(s) de sortie JSON. • **All in one block** : les données reçues sont écrites dans un bloc de données. • **One row per record** : les données reçues sont écrites dans des blocs de données ligne par ligne.
	Name of data block	Saisissez le nom du bloc de données du (des) fichier(s) de sortie JSON. *Ce champ n'est disponible que si vous sélectionnez l'option All in one block dans la liste **Output type**.*
	Action	Sélectionnez l'action que vous souhaitez effectuer sur les données : • **Overwrite** : les données dans HDFS sont écrasées si elles existent déjà. • **Create** : les données sont créées.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Dans un Job **Talend** Map/Reduce, ce composant est utilisé en tant que composant de fin et requiert un composant de transformation comme lien d'entrée. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Une fois qu'un Job Map/Reduce est ouvert dans l'espace de modélisation graphique, le **tFileOutputJSON** et toute la famille MapReduce apparaissent dans la **Palette** du Studio. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	

Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.
Prérequis	La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d'exemple relatives à MapR. • Assurez-vous d'avoir installé le client MapR sur la même machine que le *Studio Talend* et d'avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D'après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées *MAPR_INSTALL\ hadoop\hadoop-VERSION\lib\native*. Par exemple, pour Windows, la bibliothèque est *lib\MapRClient.dll* dans le fichier Jar du client MapR. Pour plus d'informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n'ajoutez pas de librairie, il est possible que vous rencontriez l'erreur suivante : `no MapRClient in java.library.path`. • Configurez l'argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]**. Cet argument fournit au studio le chemin d'accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d'utiliser entièrement l'aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR. Pour plus d'informations concernant la configuration de cet argument, consultez la section expliquant comment visualiser les données dans le *Guide de prise en main de Talend Open Studio for Big Data*. Pour plus d'informations concernant l'installation d'une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénario : Ecrire un fichier JSON structuré

Ce scénario de deux composants comprend un composant **tRowGenerator** qui génère des données au hasard, ainsi qu'un composant **tFileOutputJSON** qui écrit dans un fichier structuré JSON de sortie.

1. Glissez de la **Palette** dans l'espace de modélisation graphique un **tRowGenerator** et un **tFileOutputJSON**.
2. Liez les composants à l'aide d'un lien **Row > Main**.
3. Double-cliquez sur le **tRowGenerator** pour définir ses **Basic settings** dans la vue **Component**.

4. Cliquez sur **[...]** à côté du bouton **Edit Schema** afin d'afficher la boîte de dialogue et définir son schéma.

5. Cliquez sur le bouton **[+]** pour ajouter les colonnes désirées.

6. Sous le nœud **Columns**, saisissez le nom des colonnes.

7. Sous **Type**, sélectionnez le type dans la liste.

8. Cliquez sur **OK** pour fermer la boîte de dialogue.

9. Cliquez sur le bouton **[+]** à côté de **RowGenerator Editor** pour ouvrir la boîte de dialogue correspondante.

10. Sous **Functions**, sélectionnez les fonctions pré-définies pour les colonnes, si besoin, ou sélectionnez **[...]** pour définir les paramètres personnalisés des fonctions, dans l'onglet **Function parameters**.

11. Saisissez le nombre de lignes à générer, dans le champ **Number of Rows for RowGenerator**.

12. Cliquez sur **OK** pour fermer la boîte de dialogue.

13. Cliquez sur le composant **tFileOutputJSON** afin de définir ses propriétés, dans l'onglet **Basic settings** de la vue **Component**.

14. Cliquez sur le bouton **[...]** pour parcourir votre répertoire et définir où vous souhaitez générer le fichier de sortie JSON, puis saisissez un nom de fichier.
15. Saisissez un nom pour le bloc de données à générer dans le champ correspondant, entre guillemets doubles.
16. Sélectionnez **Built-In** dans la liste **Schema**.
17. Cliquez sur **Sync Columns** afin de retrouver le schéma du précédent composant.
18. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
19. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour exécuter le Job.

```
{
    "person":[
        {
            "birthplace":"Manchester",
            "age":48,
            "name":"Carter"
        },
        {
            "birthplace":"Liverpool",
            "age":39,
            "name":"Clinton"
        },
        {
            "birthplace":"London",
            "age":53,
            "name":"Taylor"
        }
]
}
```

Les données du schéma d'entrée sont écrites dans un bloc de données structurées JSON, dans le fichier de sortie.

tFileOutputLDIF

Propriétés du tFileOutputLDIF

Famille de composant	File/Output	
Fonction	Le composant **tFileOutputLDIF** extrait des données vers un fichier LDIF qui peut ensuite être chargé dans un annuaire LDAP.	
Objectif	Le **tFileOutputLDIF** écrit ou modifie un fichier LDIF avec des données délimitées en entrée selon le schéma défini. Peut également supprimer le contenu d'un fichier LDIF.	
Basic settings	*File name*	Chemin d'accès et nom du fichier à traiter, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Wrap	Place un retour chariot après le nombre de caractères défini.
	Change type	Sélectionnez **Add**, **Modify** ou **Delete** pour respectivement créer un nouveau fichier LDIF, modifier ou supprimer un fichier LDIF. En cas de modification, paramétrez le type de modification apportée.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Cliquez pour synchroniser le schéma du fichier de sortie avec le schéma du fichier d'entrée. La fonction **Sync** n'apparaît qu'une fois la connexion Row mise en place entre les composants d'entrée et de sortie.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
Advanced settings	*Create directory if not exists*	Cochez cette case pour créer un nouveau dossier si votre dossier de destination n'existe pas.

	Custom the flush buffer size	Cochez cette case pour personnaliser le nombre de lignes à écrire avant de vider le tampon. **Row Number** : Nombre de lignes à écrire avant de vider le tampon.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	Don't generate empty file	Cochez cette case pour empêcher la génération d'un fichier vide.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour écrire un fichier LDIF à l'aide des données provenant d'autres composants reliés par une connexion de type Row.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Ecrire des données dans un fichier LDIF

Ce scénario est constitué de deux composants et a pour objectif d'extraire les données d'une table de base de données puis d'écrire ses données dans un fichier LDIF de sortie.

Déposer et relier les composants

1. Cliquez et déposez les composants **tMysqlInput** et **tFileOutputLDIF** de la **Palette** dans l'espace de modélisation.

2. Connectez les deux composants à l'aide d'un lien **Row > Main**.

Configurer les composants

1. Sélectionnez le composant **tMysqlInput** et cliquez sur l'onglet **Basic settings** de la vue **Component**.

2. Si vous conservez les informations de connexion à votre base de données dans la zone **Metadata** du **Repository**, sélectionnez l'option **Repository** dans le champ **Property type**. De même, dans le champ **Schema type**, sélectionnez le schéma correspondant parmi les métadonnées si votre schéma est également stocké dans le **Repository**. Tous les paramètres de connexion comme de schéma sont ainsi récupérés.

3. Sinon, sélectionnez **Built-in** dans les champs **Property type** et **Schema** et définissez manuellement votre connexion à la base de données et les informations du schéma.

4. Puis double-cliquez sur le composant **tFileOutpuLDIF** et paramétrez ses propriétés dans l'onglet **Basic settings**.

5. Renseignez le nom et le chemin d'accès au fichier LDIF. S'il n'existe pas, le Job se chargera de le créer. Saisissez le nom de ce nouveau fichier.

6. Dans le champ **Wrap**, saisissez le nombre de caractères contenus sur une ligne. Les caractères qui suivent seront automatiquement placés sur la ligne suivante.

7. Sélectionnez **Add** dans le champ **Change Type** puisque, par définition, un fichier nouvellement créé est vide. Dans le cas d'une modification, vous devrez définir la nature de la modification que vous souhaitez effectuer dans le fichier.

8. Dans le champ **Schema**, sélectionnez l'option **Built-in** et cliquez sur **Sync Columns** pour récupérer la définition du schéma d'entrée.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** afin de l'exécuter.

Le fichier LDIF créé comporte les données de la table de la base de données et le type de changement effectué sur le fichier est, dans ce cas, défini comme : *addition*.

tFileOutputMSDelimited

Propriétés du tFileOutputMSDelimited

Famille du composant	File/Output	
Fonction	Le composant **tFileOutputMSDelimited** permet d'écrire des schémas multiples dans un fichier délimité.	
Objectif	Le **tFileOutputMSDelimited** crée un fichier délimité multi structure complexe à l'aide de la structure des données (schémas) venant des différents flux d'entrée de type **Row**.	
Basic settings	*File Name*	Chemin d'accès et nom du fichier de sortie, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Row Separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Field Separator	Caractère, chaîne ou expression régulière séparant les champs.
	Use Multi Field Separators	Cochez cette case attribuer un séparateur différent à chacun des schémas, à partir de la colonne **Field separator** du champ **Schemas**.
	Schemas	La table est automatiquement chargée avec des schémas résultant des différents flux d'entrée connectés au composant **tFileOutputMSDelimited**. Renseignez les relations de dépendance entre les différents schémas : **Parent row** : Saisissez le nom du flux père (en fonction du nom du lien Row qui transfère les données). **Parent key column** : Saisissez le nom de la colonne clé du flux père. **Key column** : Saisissez le nom de la colonne clé pour la ligne sélectionnée.
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	CSV options	Cochez cette case pour prendre en compte les paramètres spécifiques aux fichiers CSV, notamment la manière de protéger les caractères dans le champ **Escape char** et le type de guillemet dans le champ **Text enclosure**.
	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier délimité de sortie s'il n'existe pas déjà.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Don't generate empty file	Cochez cette case pour annuler la génération du fichier si celui-ci est vide.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour écrire un fichier délimité multi structure et séparer les champs à l'aide d'un séparateur de champ.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tFileOutputMSPositional

Propriétés du tFileOutputMSPositional

Famille du composant	File/Output	
Fonction	Le composant **tFileOutputMSPositional** permet d'écrire des schémas multiples dans un fichier positionnel.	
Objectif	Le **tFileOutputMSPositional** crée un fichier multi structure complexe à l'aide de la structure des données (schémas) venant des différents flux d'entrée de type **Row**.	
Basic settings	*File Name*	Chemin d'accès et nom du fichier à créer, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Schemas	La table est automatiquement chargée avec des schémas résultant des différents flux d'entrée connectés au composant **tFileOutputMSPositional**. Renseignez les relations de dépendance entre les différents schémas : **Parent row** : Saisissez le nom du flux père (en fonction du nom du lien Row qui transfère les données). **Parent key column** : Saisissez le nom de la colonne clé du flux père. **Key column** : Saisissez le nom de la colonne clé pour la ligne sélectionnée. **Pattern** : Saisissez le modèle (pattern) qui permet de positionner les séparateurs de champ pour chaque ligne d'entrée. **Padding char** : Saisissez entre guillemets le caractère de remplissage à utiliser. Le caractère par défaut est un espace. **Alignment** : Sélectionnez le paramètre d'alignement approprié.
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier délimité de sortie s'il n'existe pas déjà.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **Flow** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **Flow** et retourne un nombre entier.	

	NB_LINE_UNKOWN_HEADER_TYPES : nombre de lignes ayant un type d'en-tête inconnu. Cette variable est une variable **Flow** et retourne un nombre entier. **NB_LINE_PARSE_ERRORS** : nombre de lignes ayant des erreurs de parsage. Cette variable est une variable **Flow** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour écrire un fichier positionnel multi structure et séparer les champs à l'aide d'un séparateur de position.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tFileOutputMSXML

Propriétés du tFileOutputMSXML

Famille de composant	File/Output	
Fonction	Le composant **tFileOutputMSXML** permet d'écrire des schémas multiples dans un fichier XML structuré.	
Objectif	**tFileOutputMSXML** crée un fichier XML multi structure complexe à l'aide de la structure des données (schémas) venant des différents flux d'entrée de type **Row**.	
Basic settings	*File Name*	Chemin d'accès et nom du fichier à traiter, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Configure XML tree	Ouvre l'interface dédiée pour vous aider à mettre en place le mapping XML. Pour plus de détails sur l'interface, voir *Définir un arbre XML Multischéma*.
Advanced settings	*Create directory if not exists*	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier délimité de sortie s'il n'existe pas déjà.
	Advanced separator (for numbers)	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Don't generate empty file	Cochez cette case pour annuler la génération du fichier si celui-ci est vide.
	Trim the whitespace characters	Cochez cette case pour supprimer les espaces en début et en fin de colonnes.
	Escape text	Cochez cette case pour échapper les caractères spéciaux.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Définir un arbre XML Multischéma

Double-cliquez sur le composant **tFileOutputMSXML** pour ouvrir l'interface dédiée ou cliquez sur le bouton **[...]** de l'onglet **Basic settings** de la vue **Component**.

La liste déroulante de la zone **Linker source**, à gauche de l'interface, contient tous les schémas d'entrée qui peuvent être ajoutés au fichier XML multischéma de sortie (à la condition que plusieurs flux d'entrée soient connectés au composant **tFileOutputMSXML**).

Dans le champ **Schema List**, on retrouve la liste de toutes les colonnes récupérées du flux d'entrée de la sélection.

Sur la droite de l'interface, définissez la structure XML que vous souhaitez obtenir dans le fichier XML en sortie.

Vous pouvez la créer manuellement ou tout simplement importer la structure XML. Puis importez les colonnes de chaque schéma d'entrée de la sélection dans l'élément de l'arbre XML correspondant.

Importer un arbre XML

Le meilleur moyen de renseigner l'arbre XML est d'importer un fichier XML bien formé.

1. Dans la colonne **XML Tree**, cliquez une fois sur le champ **root tag**, qui s'affiche par défaut, pour le renommer.
2. Cliquez-droit sur ce champ pour afficher le menu contextuel.
3. Dans le menu, sélectionnez **Import XML tree**.

4. Sélectionnez le fichier à importer et cliquez sur **OK**.

La colonne **XML Tree** est donc automatiquement renseignée avec les bons éléments. Vous pouvez supprimer et ajouter des éléments ou des sous éléments à l'arbre :

5. Sélectionnez l'élément adéquat dans l'arbre.

6. Cliquez-droit pour faire apparaître le menu contextuel.

7. Sélectionnez **Delete** pour supprimer la sélection de l'arbre ou sélectionnez l'option adéquate parmi les suivantes : **Add sub-element**, **Add attribute**, **Add namespace** pour enrichir l'arbre.

Créer manuellement l'arbre XML

Si vous ne possédez pas de structure XML déjà définie, vous pouvez la créer manuellement.

1. Dans la colonne **XML Tree**, cliquez une fois sur le champ **root tag**, qui s'affiche par défaut, pour le renommer.

2. Cliquez-droit sur ce champ pour afficher le menu contextuel.

3. Dans le menu, sélectionnez **Add sub-element** pour créer le premier élément de la structure.

   Vous pouvez aussi ajouter un attribut ou un élément enfant à n'importe quel élément de l'arbre ou supprimer n'importe quel élément de l'arbre.

4. Sélectionnez l'élément adéquat dans l'arbre que vous venez de créer.

5. Cliquez-droit à gauche du nom de l'élément pour afficher le menu contextuel.

6. Dans le menu, sélectionnez l'option adéquate parmi les suivantes : **Add sub-element**, **Add attribute**, **Add namespace** ou **Delete**.

Mapping de données XML à partir de sources multischéma

Une fois votre arbre XML créé, sélectionnez le premier schéma à mapper.

Vous pouvez alimenter chaque élément ou sous-élément XML avec les colonnes du flux d'entrée dans la colonne **Related Column** :

1. Cliquez sur l'une des entrées de la colonne **Schema column name**.
2. Glissez-la dans le sous-élément correspondant à droite.
3. Relâchez-la pour que le mapping soit effectif.

Une flèche bleue apparaît pour illustrer ce mapping. S'il est disponible, utilisez le bouton **Auto-Map** situé en bas à gauche de l'interface, il permet d'effectuer cette opération automatiquement.

Vous pouvez déconnecter n'importe quel mapping de n'importe quel élément de l'arbre XML :

4. Sélectionnez l'élément de l'arbre XML que vous souhaitez déconnecter de sa source respective.
5. Cliquez-droit à gauche de son nom pour afficher le menu contextuel.
6. Sélectionnez **Disconnect linker**.

La flèche bleue disparaît.

Définir le statut du nœud

Définir l'arbre XML et le transfert de données ne suffit pas. Vous devez aussi définir les éléments sur lesquels la boucle est effectuée **pour chacune des sources de la sélection** et, si nécessaire, l'élément à partir duquel le regroupement est effectué.

Elément Boucle

L'élément Boucle permet de définir l'élément objet d'une itération. L'élément Boucle est généralement un générateur de lignes.

Pour définir un élément comme un élément Boucle :

1. Sélectionnez l'élément adéquat dans l'arbre XML.
2. Cliquez-droit à gauche du nom de l'élément pour afficher le menu contextuel.
3. Sélectionnez l'option **Set as Loop Element**.

La colonne **Node Status** affiche le nouveau statut sélectionné.

Il ne peut y avoir qu'un seul élément Boucle à la fois.

Elément Group

L'élément Group est optionnel, il représente un élément constant sur lequel est effectuée une fonction Groupby. L'élément Group ne peut être défini que si un élément Boucle a été préalablement défini.

Lorsque vous utilisez un élément Group, les lignes doivent être triées pour pouvoir être regroupées par le noeud sélectionné.

Pour définir un élément comme élément Group :

1. Sélectionnnez l'élément adéquat dans l'arbre XML.
2. Cliquez-droit à gauche du nom de l'élément pour afficher le menu contextuel.
3. Sélectionnez l'option **Set as Group Element**.

La colonne **Node Status** affiche le nouveau statut sélectionné et tout autre statut de regroupement est automatiquement défini, si nécessaire.

Une fois le mapping terminé, cliquez sur **OK** pour valider les paramètres et continuez la configuration de votre Job.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tFileOutputPositional

Propriétés du tFileOutputPositional

Famille de composant	File/Output	
Fonction	Le composant **tFileOutputPositional** écrit un fichier ligne par ligne en se basant sur la longueur et le format des champ ou colonnes d'une ligne. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job Map/Reduce **Talend** afin de générer du code Map/Reduce. Pour plus d'informations, consultez *Le tFileOutputPositional dans des Jobs Map/Reduce Talend*.	
Objectif	Il écrit un fichier ligne par ligne, en se basant sur la structure des données (schéma) venant du flux d'entrée.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use Output Stream	Cochez la case pour traiter le flux de données qui vous intéresse. Une fois cochée, le champ **Output Stream** s'affiche et vous pouvez saisir le flux de données souhaité. Le flux de données à traiter doit être ajouté au flux afin que ce composant récupère ces données via la variable représentative correspondante. Cette variable peut être prédéfinie dans votre Studio ou fournie par le contexte ou les composants utilisé(s) avec ce composant. Sinon, vous pouvez la définir manuellement et l'utiliser selon votre Job, par exemple à l'aide d'un **tJava** ou d'un **tJavaFlex**. Afin d'éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d'autocomplétion (**Ctrl+Espace**) afin de remplir le champ, si cette variable a été correctement définie. Pour plus d'informations concernant l'utilisation d'un flux, consultez *Scénario 2 : Lire les données d'un fichier distant en mode stream*.
	File name	Chemin d'accès et nom du fichier de sortie et/ou variable à utiliser. Ce champ est indisponible lorsque vous avez coché la case **Use Output Stream**. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**) , soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Ce composant doit fonctionner avec un **tSetDynamicSchema** afin de tirer parti de la fonctionnalité de schéma dynamique.
		Built-in : Le schéma sera créé et conservé pour ce composant. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Row separator	Saisissez le séparateur à utiliser pour identifier la fin des lignes.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Include header	Cochez cette case pour tenir compte de l'en-tête dans le fichier de sortie.
	Compress as zip file	Cochez cette case pour compresser en zip le fichier de sortie.
	Formats	Personnalisez le format des données du fichier positionnel et renseignez les colonnes du tableau Formats. **Column** : Sélectionnez la colonne que vous souhaitez personnaliser. **Size** : Saisissez la taille correspondant à la colonne. **Padding char** : Saisissez entre guillemets le caractère de remplissage à utiliser. Le caractère par défaut est un espace. **Alignment** : Sélectionnez le paramètre d'alignement approprié. **Keep** : Si les données contenues dans la colonne ou dans le champ sont trop longues, sélectionnez la partie que vous souhaitez garder.
Advanced settings	*Advanced separator (for numbers)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Use byte length as the cardinality	Cochez cette case pour permettre la prise en charge des caractères à deux octets dans ce composant. Pour cette fonction, JDK 1.6 est requis.
	Create directory if not exists	Cochez cette case pour créer un nouveau dossier si votre dossier de destination n'existe pas.
	Custom the flush buffer size	Cochez cette case pour personnaliser le nombre de lignes à écrire avant de vider le tampon. **Row Number** : Nombre de lignes à écrire avant de vider le tampon.
	Output in row mode	Cochez cette case pour que l'écriture se fasse en mode ligne.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
	Don't generate empty file	Cochez cette case pour empêcher la génération d'un fichier vide.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement	

	dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour lire un fichier et séparer les champs à l'aide du séparateur spécifié.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Le tFileOutputPositional dans des Jobs Map/Reduce Talend

*Les informations contenues dans cette section concernent uniquement les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend** et ne sont pas applicables aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job Map/Reduce **Talend**, le **tFileOutputPositional**, ainsi que le Job complet utilisant ce composant, génère du code Map/Reduce natif. Cette section présente les propriétés spécifiques du **tFileOutputPositional** lorsqu'il est utilisé dans un Job Map/Reduce. Pour plus d'informations concernant un Job Map/Reduce **Talend**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	MapReduce/Output	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Réutilisez les propriétés stockées centralement sous le nœud **Hadoop Cluster** de la vue **Repository**. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Pour plus d'informations concernant le nœud **Hadoop Cluster**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.
		Cliquez sur cette icône pour ouvrir l'assistant de connexion à la base de données et stocker les paramètres de connexion configurés dans la vue **Basic settings**. Pour plus d'informations concernant la configuration et le stockage des paramètres de connexion à la base de données, consultez le *Guide utilisateur du* Studio Talend.

	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Folder	Parcourez votre système ou saisissez le chemin d'accès au répertoire de HDFS dans lequel sont stockées les données à utiliser. Ce chemin d'accès doit pointer vers un dossier et non un fichier car un Job Map/Reduce **Talend** écrit dans le dossier cible le résultat final ainsi que les différents fichiers *part-* générés lors des calculs Map/Reduce. Assurez-vous que la connexion à la distribution Hadoop utilisée est correctement configurée dans l'onglet **Hadoop configuration** de la vue **Run**.
	Action	Sélectionnez une opération pour créer les données : **Create** : crée un fichier et écrit des données dans ce fichier. **Overwrite** : écrase le fichier existant dans le répertoire spécifié dans le champ **Folder**.
	Compress the data	Cochez la case **Compress the data** afin de compresser les données de sortie. Hadoop fournit différents formats de compression permettant de réduire l'espace nécessaire au stockage des fichiers et d'accélérer le transfert de données. Lorsque vous lisez un fichier compressé, le *Studio Talend* doit le décompresser avant de pouvoir en alimente le flux d'entrée.
	Formats	Personnalisez le format de données du fichier positionnel et renseignez les colonnes de la table **Formats**. **Column** : sélectionnez la colonne à personnaliser. **Size** : saisissez la taille de la colonne. **Padding char** : saisissez entre guillemets les caractères de remplissage utilisés, un espace par défaut. **Alignment** : sélectionnez le paramètre d'alignement approprié. **Keep** : si les données dans la colonne ou dans le champ sont trop longues, sélectionnez la partie à conserver.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Dans un Job **Talend** Map/Reduce, ce composant est utilisé en tant que composant de fin et requiert un composant de transformation comme lien d'entrée. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Lorsqu'un Job Map/Reduce est ouvert dans l'espace de modélisation graphique, le **tFileOutputDelimited** et la famille **MapReduce** s'affichent dans la **Palette** du Studio. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.

Scénario associé

Pour un scénario associé, consultez *Scénario : Transformer en fichier Regex en Positional*.

Pour un scénario relatif à l'utilisation de l'option **Use Output Stream**, consultez *Scénario 2 : Utiliser un flux de sortie pour sauvegarder des données dans un fichier local*.

tFileOutputProperties

Propriétés du tFileOutputProperties

Famille de composant	File/Output	
Fonction	Le composant **tFileInputProperties** écrit un fichier de configuration de type ini ou properties.	
Objectif	Il écrit un fichier de configuration contenant des données texte organisées sur le modèle clé=valeur.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Il se compose de deux colonnes, *Key* et *Value*, correspondant au nom et à la valeur du paramètre à copier.
	File format	Sélectionnez le format de votre fichier **.properties** ou **.ini**.
		.properties : fichier de configuration dont les données sont disposées sur deux lignes et structurées de la manière suivante clé=valeur.
		.ini : fichier de configuration dont les données sont disposées sur deux lignes, structurées de la manière suivante clé=valeur. et regroupées en sections.. **Section Name** : saisissez le nom de la section sur laquelle effectuer l'opération d'itération.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	File Name	Chemin d'accès et nom du fichier de sortie, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
Advanced settings	*Encoding*	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour écrire un fichier dont les données sont organisées selon la structure clé=valeur.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Lire et mapper la clé et les valeurs de fichiers properties et alimenter un glossaire* de la section *tFileInputProperties*.

tFileOutputRaw

Propriétés du tFileOutputRaw

Famille de composant	File/Output	
Fonction	Le composant **tFileOutputRaw** gère les données de sortie dans un Job fournissant une seule colonne de données de sortie.	
Objectif	Ce composant reçoit des données fournies par un autre composant en une seule colonne.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Filename	Nom et chemin d'accès au fichier d'entrée à traiter. Vous pouvez saisir le chemin manuellement entre guillemets doubles ou parcourir votre système en cliquant sur le bouton **[...]**.
	Encoding	Si cette sortie est une chaîne de caractères, sélectionnez le type d'encodage dans la liste ou sélectionnez **Custom** et définissez-le manuellement.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**. Pour capturer l'exception `FileNotFoundException`, vous devez cocher cette case.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**FILENAME_PATH** : chemin du fichier d'entrée. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Le composant **tFileOutputRaw** est utilisé pour recevoir des données provenant d'une source de données fournissant ces données en une seule colonne.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tFileOutputXML

Le **tFileOutputXML** appartient à deux familles de composants : File et XML. Pour plus d'informations concernant le composant **tFileOutputXML**, consultez *tFileOutputXML*.

tFileProperties

Propriétés du tFileProperties

Famille du composant	File/Management	
Fonction	Le composant **tFileProperties** permet de créer un flux monoligne affichant les propriétés du fichier traité.	
Objectif	Le **tFileProperties** permet de recueillir les informations concernant les propriétés principales d'un fichier défini.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit les champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Il décrit les propriétés principales du fichier spécifié. Vous pouvez cliquer sur le bouton **[...]** à côté de **Edit schema** pour voir le schéma prédéfini qui contient les champs suivants : • **abs_path** : le chemin absolu du fichier. • **dirname** : le répertoire du fichier. • **basename** : le nom du fichier. • **mode_string** : le mode d'accès du fichier, *r* et *w* pour les permissions de lecture et d'écriture respectivement. • **size** : la taille du fichier en octets. • **mtime** : le timestamp indiquant quand le fichier a été modifié pour la dernière fois, en millisecondes qui se sont écoulées depuis l'époque UNIX (1er janvier 1970, 00:00:00 UTC). • **mtime_string** : la date et l'heure auxquelles le fichier a été modifié pour la dernière fois.
	File	Chemin d'accès et nom du fichier à traiter, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Calculate MD5 Hash	Cochez cette case pour vérifier le MD5 du fichier téléchargé.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en standalone.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur de Studio Talend*.	

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Afficher les propriétés d'un fichier traité

Ce scénario décrit un Job très simple permettant d'afficher les propriétés d'un fichier spécifié.

1. A partir de la **Palette**, cliquez-déposez les composants **tFileProperties** et **tLogRow** dans l'éditeur graphique.
2. Cliquez-droit sur le **tFileProperties** et connectez-le au **tLogRow** à l'aide d'un lien de type **Row Main**.

3. Dans l'éditeur graphique, sélectionnez le composant **tFileProperties**.
4. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tFileProperties**.

5. Configurez le champ **Schema** en mode **Built-In**.
6. Cliquez sur le bouton **[...]** du champ **Edit schema** si vous voulez voir les colonnes en lecture seule.
7. Dans le champ **File**, saisissez le chemin d'accès ou parcourez vos dossiers jusqu'au fichier dont vous voulez afficher les propriétés.
8. Dans l'éditeur graphique, sélectionnez le **tLogRow** et cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**). Pour plus d'informations, consultez *tLogRow*.

9. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

```
Starting job File_Properties at 14:57 26/08/2008.
.-----------------------------------------------.
|                #1. tLogRow_1                  |
+---------------+-------------------------------+
| key           | value                         |
+---------------+-------------------------------+
| abs_path      | C:\test5.txt                  |
| dirname       | C:\                           |
| basename      | test5.txt                     |
| mode_string   | rw                            |
| size          | 86                            |
| mtime         | 1219674736421                 |
| mtime_string  | Mon Aug 25 16:32:16 CEST 2008 |
+---------------+-------------------------------+

Job File_Properties ended at 14:57 26/08/2008. [exit code=0]
```

Les propriétés du fichier défini s'affichent dans la console **Run**.

tFileRowCount

Propriétés du tFileRowCount

Famille de composant	File/Management	
Fonction	Le composant **tFileRowCount** compte le nombre de lignes dans un fichier.	
Objectif	Ce composant ouvre un fichier et le lit ligne par ligne afin de déterminer le nombre de lignes qu'il contient.	
Basic settings	*File Name*	Chemin d'accès et nom du fichier à traiter, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Ignore empty rows	Cochez cette case pour que les lignes vides ne soient pas prises en compte.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Le **tFileRowCount** est un composant standalone, et doit être utilisé avec un lien de type **OnSubjobOk** vers un **tJava**.	
Global Variables	**COUNT** : nombre de lignes dans un fichier. Cette variable est une variable **Flow** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : OnSubjobOk, OnSubjobError, Run if. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez la section relatives aux types de connexions, dans le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Ecrire un fichier dans MySQL si le nombre d'enregistrements correspond à une valeur de référence

Dans ce scénario, le **tFileRowCount** compte le nombre d'enregistrements d'un fichier .txt et compare ce nombre à une valeur de référence grâce au **tJava**. Lorsque les deux valeurs correspondent, le fichier .txt est écrit dans une table MySQL.

Le fichier .txt contient deux enregistrements :

```
1;andy
2;mike
```

Relier les composants

1. Déposez un **tFileRowCount**, un **tJava**, un **tFileInputDelimited** et un **tMysqlOutput** de la **Palette** dans l'espace de modélisation graphique.
2. Reliez le **tFileRowCount** au **tJava** à l'aide d'un lien **Trigger** > **OnSubjobOk**.
3. Reliez le **tJava** au **tFileInputDelimited** à l'aide d'un lien **Trigger** > **Run if**.
4. Reliez le **tFileInputDelimited** au **tMysqlOutput** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFileRowCount** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **File Name**, saisissez le chemin complet vers le fichier .txt. Vous pouvez également cliquer sur le bouton pour parcourir votre système jusqu'au fichier.

   Cochez ensuite la case **Ignore empty rows**.

3. Double-cliquez sur le **tJava** pour ouvrir sa vue **Basic settings**.

   Dans la zone **Code**, saisissez la fonction permettant d'afficher le nombre de lignes du fichier :

```
System.out.println(globalMap.get("tFileRowCount_1_COUNT"));
```

4. Cliquez sur le lien **If** pour ouvrir sa vue **Basic settings**.

   Dans la zone **Condition**, saisissez l'instruction permettant de déterminer si le nombre de lignes est égal à *2* :

```
((Integer)globalMap.get("tFileRowCount_1_COUNT"))==2
```

   Ce lien **If** signifie que si le nombre de lignes est égal à *2*, les lignes du fichier .txt sont écrites dans MySQL.

5. Double-cliquez sur le **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.

Dans le champ **File name/Stream**, saisissez le chemin complet vers le fichier .txt. Vous pouvez également cliquer sur le bouton afin de parcourir votre système jusqu'au fichier.

6. Cliquez sur le bouton **Edit schema** afin d'ouvrir l'éditeur de schéma.

7. Cliquez deux fois sur le bouton afin d'ajouter deux colonnes, nommées respectivement *id* (de type *Integer*) et *name* (de type *String*).

8. Dans la boîte de dialogue qui s'affiche, cliquez sur le bouton **Yes** afin de propager le schéma au composant suivant.

9. Double-cliquez sur le **tMysqlOutput** pour ouvrir sa vue **Basic settings**.

10. Dans les champs **Host** et **Port**, saisissez les informations de connexion.

    Dans le champ **Database**, saisissez le nom de la base de données.

    Dans les champs **Username** et **Password**, saisissez les informations d'authentification.

    Dans le champ **Table**, saisissez le nom de la table, par exemple *"staff"*.

    Dans la liste **Action on table**, sélectionnez **Create a table if not exists**.

    Dans la liste **Action on data**, sélectionnez **Insert**.

Exécuter le Job

1. Appuyez sur **Ctrl+S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** pour exécuter le Job.

```
Starting job tFileRowCount at 10:35 27/08/2013.

[statistics] connecting to socket on port 3610
[statistics] connected
2
[statistics] disconnected
Job tFileRowCount ended at 10:35 27/08/2013. [exit code=0]
```

   Comme montré ci-dessus, le Job a été exécuté avec succès et le nombre de lignes du fichier .txt a été affiché.

3. Ouvrez l'interface graphique de MySQL et ouvrez la table *staff*.

Comme montré ci-dessus, la table a été créée et deux enregistrements ont été insérés.

tFileTouch

Propriétés du tFileTouch

Famille de composant	File/Management	
Fonction	Le composant **tFileTouch** crée un fichier vide ou, si le fichier spécifié existe déjà, met à jour sa date de modification et de dernier accès et laisse son contenu inchangé.	
Objectif	Ce composant crée un fichier vide ou modifie les détails d'un fichier existant, et crée le répertoire de destination s'il n'existe pas.	
Basic settings	*File Name*	Chemin d'accès et nom du fichier à créer, et/ou variable à utiliser.
	Create directory if not exists	Cochez cette case pour créer un nouveau dossier si votre dossier de destination n'existe pas.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en standalone.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tFileUnarchive

Propriétés du tFileUnarchive

Famille de composant	File/Management	
Fonction	Le composant **tFileUnarchive** dézippe un fichier archive en fonction des paramètres définis et le place dans le répertoire souhaité.	
Objectif	Ce composant dézippe un fichier archive de n'importe quel format (les formats *.tar.gz, *.tgz, *.tar, *.gz et *.zip sont supportés) pour un éventuel traitement.	
Basic settings	*Archive file*	Chemin d'accès au fichier archive.
	Extraction Directory	Répertoire dans lequel le fichier dézippé sera placé.
	Use archive name as root directory	Cochez cette case pour créer un dossier du même nom que l'archive, s'il n'existe pas, dans le répertoire spécifié et extraire le(s) fichier(s) dézippé(s) dans ce dossier.
	Check the integrity before unzip	Cochez cette case pour effectuer une vérification d'intégrité avant de dézipper l'archive.
	Extract file paths	Cochez cette case pour reproduire la structure du chemin d'accès au fichier zippé dans l'archive.
	Need a password	Cochez cette case et saisissez le mot de passe correct si la décompression de l'archive est protégée par un mot de passe. Notez que l'archive cryptée doit être créée par le composant **tFileArchive**, sinon des messages d'erreur apparaîtront ou rien ne sera extrait, même si aucun message d'erreur n'est apparu. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en standalone mais il peut aussi être utilisé en composant de départ dans un Job via un lien **Iterate**.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate. **Trigger** : Run if, OnSubjobOk, OnSubjobError, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.	
Global Variables	**CURRENT_FILE** : nom du fichier courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_FILEPATH** : nom du fichier courant ainsi que son chemin d'accès. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	*Les fichiers aux formats suivants peuvent être décompressés : *.tar.gz , *.tgz, *.tar, *.gz et *.zip.*

Scénario associé

Pour un exemple d'utilisation du composant **tFileUnarchive**, consultez *tFileCompare*.

tGPGDecrypt

Propriétés du tGPGDecrypt

Famille de conposant	File/Management	
Fonction	Le composant **tGPGDecrypt** décrypte un fichier crypté GnuPG et sauvegarde le fichier décrypté dans le répertoire cible spécifié.	
Objectif	Ce composant appelle la commande `gpg -d` afin de décrypter un fichier crypté GnuPG et sauvegarde le fichier décrypté dans le répertoire spécifié.	
Basic settings	*Input encrypted file*	Chemin d'accès au fichier crypté file.
	Output decrypted file	Chemin d'accès au fichier décrypté.
	GPG binary path	Chemin d'accès à la commande GPG.
	Passphrase	Saisissez la phrase secrète utilisée pour crypter le fichier d'entrée spécifié. Pour saisir la phrase secrète, cliquez sur le bouton **[...]** à côté du champ **Passphrase**, puis, dans la boîte de dialogue qui s'ouvre, saisissez la phrase secrète entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
	No TTY Terminal	Cochez cette case pour spécifier qu'aucun terminal TTY n'est utilisé en ajoutant l'option `--no-tty` à la commande de décryptage.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**FILE** : nom du fichier de sortie. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **FILEPATH** : chemin d'accès au fichier de sortie. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone.	

Scénario : Décrypter un fichier crypté GnuPG et afficher son contenu

Le scénario suivant décrit un Job de trois composants qui décrypte un fichier crypté GnuPG et affiche son contenu dans la console de la vue **Run**.

Déposer et relier les composants

1. Déposez un composant **tGPGDecrypt**, un **tFileInputDelimited** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.

2. Reliez le composant **tGPGDecrypt** au **tFileInputDelimited** à l'aide d'un lien **Trigger** > **OnSubjobOk** puis connectez le **tFileInputDelimited** au **tLogRow** en utilisant un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le composant **tGPGDecrypt** afin d'ouvrir sa vue **Component** et définir ses propriétés :

2. Dans le champ **Input encrypted file**, parcourez votre répertoire jusqu'au fichier à décrypter.

3. Dans le champ **Output decrypted file**, saisissez le chemin d'accès au fichier décrypté.

   Si le chemin d'accès du fichier contient des caractères accentués, vous obtiendrez un message d'erreur lors de l'exécution du Job. Pour plus d'informations concernant la procédure à suivre lorsque les accents ne sont pas supportés, consultez le Guide d'installation et de migration Talend.

4. Dans le champ **GPG binary path**, parcourez votre répertoire jusqu'au fichier de commandes GPG.

5. Dans le champ **Passphrase**, saisissez la phrase secrète utilisée lors du cryptage du fichier d'entrée.

6. Double-cliquez sur le composant **tFileInputDelimited** afin d'ouvrir sa vue **Component** et définir ses propriétés :

7. Dans la liste **Property Type**, sélectionnez **Built-In**.

8. Dans le champ **File name/Stream**, définissez le chemin d'accès au fichier décrypté, le chemin de sortie défini dans votre composant **tGPGDecrypt**.

9. Dans le champ **Header**, définissez le nombre de lignes à ignorer au début du fichier. Dans cet exemple, saisissez *1*.

10. Dans le champ **Footer**, saisissez le nombre de lignes à ignorer à la fin du fichier. Dans cet exemple, aucune ligne n'est à ignorer, saisissez donc *0*.

11. Dans le champ **Limit**, saisissez le nombre de lignes à traiter. Saisissez *10*.

12. Dans la liste **Schema**, sélectionnez **Built-In**. Cela signifie que les propriétés que vous configurez dans le composant ne seront pas réutilisables dans d'autres Jobs.

13. Cliquez sur **Edit schema** et éditez le schéma du composant. Cliquez deux fois sur le bouton **[+]** pour ajouter deux colonnes que vous nommerez *idState* et *labelState*.

14. Cliquez sur **OK** afin de valider vos modifications et fermer l'éditeur.

15. Double-cliquez sur le composant **tLogRow** afin d'afficher sa vue **Component** et définir ses propriétés.

16. Dans la liste **Schema**, sélectionnez **Built-In**.

17. Dans la zone **Mode**, sélectionnez l'option **Table (print values in cells of a table)**.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

```
Starting job tGPGDecrypt at 18:49
17/11/2010.

[statistics] connecting to socket on port
3990
[statistics] connected
.-------+----------.
|     tLogRow_1     |
|=------+----------=|
|idState|labelState |
|=------+----------=|
|1      |Alabama    |
|2      |Alaska     |
|3      |Arizona    |
|4      |Arkansas   |
|5      |California |
|6      |Colorado   |
|7      |Connecticut|
|8      |Delaware   |
|9      |Florida    |
|10     |Georgia    |
`-------+----------'

[statistics] disconnected
Job tGPGDecrypt ended at 18:49 17/11/2010.
[exit code=0]
```

Le fichier spécifié est décrypté et le nombre défini de lignes s'affiche dans la console de la vue **Run**.

tNamedPipeClose

Propriétés du tNamedPipeClose

Famille de composant	File/Input	
Fonction	Le composant **tNamedPipeClose** ferme un tube nommé ouvert par le **tNamedPipeOpen**, à la fin d'un processus.	
Objectif	Ce composant est utilisé pour fermé un tube nommé à la fin d'un processus.	
Basic settings	*Pipe*	Sélectionnez dans la liste un tube nommé existant à fermer.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion au tube nommé parmi celles prévues dans votre Job. Lorsqu'un paramètre dynamique est configuré, la case **Pipe** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé pour fermer un tube nommé à la fin d'un Job.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Ecrire et charger des données via un tube nommé*.

tNamedPipeOpen

Propriétés du tNamedPipeOpen

Famille de composant	File/Input	
Fonction	Le composant **tNamedPipeOpen** ouvre un tube nommé pour écrire des données à l'intérieur.	
Objectif	Ce composant est utilisé dans un processus de communication interne, il ouvre un tube nommé pour écrire des données à l'intérieur.	
Basic settings	*Name*	Saisissez dans le champ le nom du tube nommé.
	Delete if already exist	Cochez cette case si vous ne souhaitez pas dupliquer un tube nommé. Le tube nommé existant sera remplacé.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**PIPE_NAME** : nom du tube nommé. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **PIPE_NATIVE_NAME** : nom natif du tube nommé. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **PIPE_OUTPUTSTREAM** : flux de sortie du tube nommé. Cette variable est une variable **Flow** et retourne un object. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début dans un Job effectuant un processus de communication interne.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Ecrire et charger des données via un tube nommé*.

tNamedPipeOutput

Propriétés du tNamedPipeOutput

Famille de composant	File/Input	
Fonction	Le composant **tNamedPipeOutput** écrit des données dans un tube nommé ouvert.	
Objectif	Ce composant vous permet d'écrire des données dans un tube nommé ouvert déjà existant.	
Basic settings	*Use existing pipe connection*	Cochez cette case afin d'utiliser un tube nommé existant dans la liste **Pipe component**, ou décochez cette case afin de spécifier un tube nommé dans le champ **Pipe name**.
	Pipe component	Sélectionnez dans la liste un composant NamedPipe existant. Cette case est disponible uniquement lorsque vous cochez la case **Use existing pipe connection**.
	Pipe name	Saisissez dans le champ le nom d'un tube nommé existant. Cette case s'affiche uniquement lorsque vous décochez la case **Use existing pipe connection**.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes dans le fichier de sortie.
	Field separator	Caractère, chaîne ou expression régulière pour séparer les champs du fichier de sortie.
	CSV options	Cochez cette case afin de prendre en compte tous les paramètres spécifiques aux fichiers CSV, en particulier les paramètres **Escape char** et **Text enclosure**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Delete pipe if it exists	Cochez cette case si vous ne souhaitez pas dupliquer un tube nommé. Le tube nommé existant sera remplacé.

Advanced settings	*Boolean type*	Sélectionnez un type de booléen dans la liste.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion au tube nommé parmi celles prévues dans votre Job. La table **Dynamic settings** n'est disponible que si la case **Use existing pipe connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Pipe component** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **Flow** et retourne un entier. **PIPE_NAME** : nom du tube nommé. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **PIPE_NATIVE_NAME** : nom natif du tube nommé. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **PIPE_OUTPUTSTREAM** : flux de sortie du tube nommé. Cette variable est une variable **Flow** et retourne un object. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé connecté à un autre composant dans un sous-job lisant des données d'une source.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Ecrire et charger des données via un tube nommé

Le scénario suivant permet de créer un Job écrivant des données dans un tube nommé ouvert et affichant ces données dans la console.

Déposer et relier les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tNamedPipeOpen**, **tParallelize**, **tNamedPipeClose**, **tFileInputDelimited**, **tSleep**, **tLogRow**, **tRowGenerator** et **tNamedPipeOutput**.

2. Connectez le **tNamedPipeOpen** au **tParallelize** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

3. Reliez le composant **tParallelize** au **tFileInputDelimited** à l'aide d'un lien **Trigger** > **Parallelize**.

4. Connectez le **tParallelize** au composant **tSleep** à l'aide d'un lien **Trigger** > **Parallelize**.

5. Reliez le **tFileInputDelimited** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

6. Connectez le composant **tParallelize** au **tNamedPipeClose** à l'aide d'un lien **Trigger** > **Synchronize (Wait for all)**.

7. Reliez le **tSleep** au composant **tRowGenerator** à l'aide d'un lien **Trigger** > **OnComponentOk**.

8. Connectez le **tRowGenerator** au **tNamedPipeOutput** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le composant **tNamedPipeOpen** afin d'ouvrir sa vue **Basic settings**.

   Dans le champ **Name**, saisissez le nom d'un tube nommé et sélectionnez **Delete if already exist** afin de ne pas dupliquer ce tube nommé.

2. Double-cliquez sur le **tParallelize** afin d'afficher sa vue **Basic settings**.

   Sélectionnez **end of all subjobs** dans la liste **Wait for**.

   Dans le champ **Sleep Duration**, saisissez *100* pour configurer la durée de pause.

3. Double-cliquez sur le **tFileInputDelimited** afin d'afficher sa vue **Basic settings**.

   Dans le champ **File name/Stream**, saisissez l'expression suivante pour utiliser le nom du tube nommé existant défini dans la vue **Basic settings** du **tNamedPipeOpen** :

```
((String)globalMap.get("tNamedPipeOpen_1_PIPE_NATIVE_NAME"))
```

4. Cliquez sur le bouton **[...]** à côté du champ **Edit schema**.

   Cliquez sur le bouton **[+]** pour ajouter trois colonnes au **tFileInputDelimited**. Renommez-les *id*, *first_name* et *last_name* puis configurez le **Type** de la colonne *id* à **Integer**. Laissez les autres paramètres tels qu'ils sont.

   Cliquez sur **OK** afin de sauvegarder le schéma.

5. Laissez les autres paramètres de la vue **Basic settings** tels qu'ils sont.
6. Double-cliquez sur le **tSleep** et saisissez *1* dans le champ **Pause (in seconds)**.
7. Double-cliquez sur le **tRowGenerator** pour afficher sa vue **Basic settings**.
8. Cliquez sur **RowGenerator Editor** pour configurer le schéma.
9. Cliquez sur le bouton **[+]** pour ajouter trois colonnes au **tRowGenerator**. Renommez ces trois colonnes respectivement *id*, *first_name* et *last_name* puis configurez le **Type** de la colonne *id* à **Integer**. Laissez les autres types tels qu'ils sont par défaut.

10. Sélectionnez **sequence** dans la liste **Functions** pour la colonne *id*.
11. Sélectionnez **getFirstName** dans la liste **Functions** pour la colonne *first_name*.
12. Sélectionnez **TalendDataGenerator.getLastName** dans la liste **Functions** pour la colonne *last_name*.
13. Sélectionnez *id*, saisissez *s1* dans le champ **Value** sous l'onglet **Function parameters** pour **sequence identifier**, *1001* pour **start value** et *1* pour **step**.

14. Cliquez sur **OK** pour sauvegarder les modifications.

15. Double-cliquez sur le composant **tNamedPipeOutput** pour afficher sa vue **Basic settings**.

    Cochez la case **Use existing pipe connection** et sélectionnez **tNamedPipeOpen_1** dans la liste **Pipe component**.

    Cochez la case **Delete pipe if it exists** pour ne pas dupliquer le tube nommé.

    Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.

    Laissez les autres paramètres tels qu'ils sont.

16. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.

17. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.

    Dans la zone **Mode**, sélectionnez **Table**.

18. Double-cliquez sur le **tNamedPipeClose** afin d'afficher sa vue **Basic settings**.

19. Sélectionnez le composant **tNamedPipeOpen_1** dans la liste **Pipe**.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

```
Starting job namedPipe_Sample_01 at 18:13 19/05/2011.

[statistics] connecting to socket on port 3842
[statistics] connected
.----+----------+---------.
|        tLogRow_1        |
|=----+----------+--------=|
|id   |first_name|last_name|
|=----+----------+--------=|
|1001|Bill      |Cleveland|
'----+----------+---------'
[statistics] disconnected
Job namedPipe_Sample_01 ended at 18:13 19/05/2011.
[exit code=0]
```

Les données écrites dans le tube nommé sont affichées dans la console.

tPivotToColumnsDelimited

Propriétés du tPivotToColumnsDelimited

Famille de composant	File/Output	
Fonction	Le composant **tPivotToColumnsDelimited** extrait les données en fonction d'une opération d'agrégation exercée sur une colonne pivot.	
Objectif	Le **tPivotToColumnsDelimited** permet d'ajuster la sélection de données à transmettre en sortie.	
Basic settings	*Pivot column*	Sélectionnez la colonne qui sert de pivot pour l'opération d'agrégation dans le flux d'entrée.
	Aggregation column	Sélectionnez la colonne contenant les données à agréger dans le flux d'entrée.
	Aggregation function	Sélectionnez la fonction à utiliser dans le cas où plusieurs valeurs sont disponibles dans la colonne pivot.
	Group by	Définissez les ensembles d'agrégation dont les valeurs qui sont utilisées pour les calculs.
		Input Column : Faites le lien entre les libellés des colonnes d'entrée et ceux des colonnes de sortie, dans le cas où vous souhaitez que les libellés du schéma de sortie soient différents du schéma d'entrée.
	File Name	Chemin d'accès et nom du fichier de sortie, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Field separator	Caractère, chaîne ou expression régulière pour séparer les champs du fichier de sortie.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
Global Variables	**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_OUT** : nombre de lignes écrites dans le fichier par le composant. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant nécessite un flux d'entrée.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external	

	+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Utiliser une colonne pivot pour agréger des données

Le scénario suivant décrit un Job qui agrège des données d'un fichier d'entrée délimité à l'aide d'une colonne pivot.

Déposer et relier les composants

1. Cliquez sur les composants suivant et déposez-les dans l'espace de modélisation : **tFileInputDelimited**, **tPivotToColumnsDelimited**.

2. Reliez les composants à l'aide d'un lien **Row** > **Main**.

Configurer les composants

Paramétrer le composant d'entrée

1. Double-cliquez sur le **tFileInputDelimited** pour afficher sa vue **Basic settings**.

2. Parcourez votre système jusqu'au fichier d'entrée dans le champ **File Name**.

   Le fichier à utiliser se compose de trois colonnes, *ID*, *Question* et *Answer*.

Id	Question	Answer
1	Name	Juan
2	Name	Jean
3	Name	John
1	Gender	M
2	Gender	F
3	Gender	M
1	Surgery	Yes
2	Surgery	No
3	Surgery	Yes
1	Age	45
2	Age	23
3	Age	42
1	Name	Mary

3. Définissez les séparateurs de lignes (**Row**) et de champs (**Field**), respectivement : retour chariot et point virgule, dans cet exemple.
4. Le fichier contient un en-tête, ainsi définissez-le.
5. Paramétrez le schéma avec les trois colonnes : *ID*, *Questions*, *Answers*.

Paramétrer le composant de sortie

1. Puis sélectionnez le composant **tPivotToColumnsDelimited** et paramétrez ses propriétés dans l'onglet **Basic settings** de la vue **Component**.

2. Dans le champ **Pivot column**, sélectionnez la colonne pivot dans le schéma d'entrée. La colonne pivot est généralement celle présentant le plus de doublons (valeurs d'agrégation pivot).
3. Dans le champ **Aggregation column**, sélectionnez la colonne à agréger dans le schéma d'entrée.
4. Dans le champ **Aggregation function**, sélectionnez la fonction à effectuer lorsque des doublons sont trouvés.
5. Dans la table **Group by**, ajoutez une entrée dans la colonne **Input column**, qui sera utilisée pour regrouper la colonne agrégée.
6. Dans le champ **File Name**, indiquez le chemin d'accès et le nom du fichier de sortie. Et dans les champs **Row** et **Field separator**, paramétrez les séparateurs pour les lignes et données de sortie ayant été agrégées.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

id	Name	Gender	Surgery	Age
1	Mary	M	Yes	45
2	Jean	M	No	23
3	John	F	Yes	42

Le fichier de sortie affiche les données qui viennent d'être agrégées.

Composants Internet

Ce chapitre passe en revue les propriétés des principaux composants présents dans la famille **Internet** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **Internet** regroupe les composants vous permettant d'accéder à du contenu stocké sur Internet, des services Web aux e-mails en passant par les flux RSS, le SCP, le MOM, le FTP, etc.

tFileFetch

Propriétés du tFileFetch

Famille de composant	Internet	
Fonction	Le composant **tFileFetch** récupère un fichier à partir d'un protocole.	
Objectif	Ce composant permet de récupérer des données d'un fichier à partir d'un protocole.	
Basic settings	*Protocol*	Sélectionnez parmi la liste suivante le protocole à utiliser, puis renseignez les champs correspondants : **http**, **https**, **ftp**, **smb**. Les propriétés peuvent différer légèrement en fonction du type de protocole sélectionné. Les champs additionnels sont définis dans ce tableau à la suite des propriétés générales.
	URI	Saisissez l'URI du site sur lequel le fichier doit être récupéré.
	Use cache to save the resource	Cochez cette case pour enregistrer les données dans le cache. Cette option vous permet de traiter directement le flux de données du fichier (en mode stream) sans sauvegarder ce fichier sur votre disque. Elle permet donc un gain de temps et de performance.
	Domain	Saisissez le nom du domaine du serveur Microsoft. Disponible pour le protocole **smb**.
	Username et *Password*	Saisissez l'identifiant utilisateur et le mot de passe de connexion au serveur. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Disponible pour le protocole **smb**.
	Destination directory	Sélectionnez le répertoire de destination dans lequel le fichier récupéré sera placé.
	Destination Filename	Saisissez le nouveau nom du fichier récupéré.
	Create full path according to URI	Cette case permet de reproduire l'arborescence de répertoires de l'URI. Pour enregistrer le fichier à la racine de votre répertoire de destination, décochez la case. Disponible pour les protocoles **http**, **https** et **ftp**.
	Add header	Cochez cette case si vous souhaitez ajouter un ou plusieurs en-tête(s) comme conditions de récupération. Dans le tableau **Headers**, saisissez le(s) paramètre(s) d'en-tête HTTP dans le champ **Name** et la valeur correspondante dans le champ **Value**. Disponible pour les protocoles **http** et **https**.
	POST method	Cette case est cochée par défaut, elle permet d'utiliser la méthode POST. Dans le tableau **Parameters**, saisissez le nom de la ou des variable(s) dans le champ **Name** et leur valeur correspondante dans le champ **Value**. Décochez cette case si vous souhaitez utiliser la méthode GET. Disponible pour les protocoles **http** et **https**.
	Die on error	Décochez cette case pour passer les lignes en erreur et terminer le traitement des lignes sans erreur.

		Disponible pour les protocoles **http**, **https** et **ftp**.
	Read Cookie	Cochez cette case pour que le **tFileFetch** charge un cookie d'authentification d'une page Web. Disponible pour les protocoles **http**, **https**, **ftp** et **smb**.
	Save Cookie	Cochez cette case pour sauvegarder le cookie d'authentification Web. Cela signifie que vous n'aurez plus à vous identifier sur le même site Web. Disponible pour les protocoles **http**, **https**, **ftp** et **smb**.
	Cookie file	Saisissez le chemin d'accès complet au fichier à utiliser pour sauvegarder les cookies. Cliquez sur le bouton **[...]** et parcourez votre système jusqu'au fichier dans lequel sauvegarder votre cookie. Disponible pour les protocoles **http**, **https**, **ftp** et **smb**.
	Cookie policy	Sélectionnez une politique relative aux cookies dans la liste déroulante. Quatre options sont disponibles : **BROWSER_COMPATIBILITY**, **DEFAULT**, **NETSCAPE** et **RFC_2109**. Disponible pour les protocoles **http**, **https**, **ftp** et **smb**.
	Single cookie header	Cochez cette case afin de rassembler tous les cookies dans l'en-tête de la requête afin d'obtenir une compatibilité maximum entre les différents serveurs. Disponible pour les protocoles **http**, **https**, **ftp** et **smb**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
	Timeout	Nombre de millisecondes avant fermeture de la connexion au protocole. Disponible pour les protocoles **http** et **https**.
	Print response to console	Cochez cette case pour imprimer la réponse du serveur dans la console. Disponible pour les protocoles **http** et **https**.
	Upload file	Cochez cette case pour charger un ou plusieurs fichier(s) sur le serveur. Puis, dans la table **Files** qui s'affiche, cliquez sur le bouton **[+]** afin d'ajouter le (les) fichier(s) à charger et définissez les paramètres suivant pour chaque fichier : • **Name** : le nouveau du fichier après chargement, entre guillemets doubles. • **File** : le chemin complet du fichier à charger, par exemple *"D:/filefetch.txt"*. • **Content-Type** : le type de contenu du fichier à charger. La valeur par défaut est *"application/octet-stream"*. • **Charset** : l'ensemble de caractères du fichier à charger. La valeur par défaut est *"ISO-8859-1"*. Disponible pour les protocoles **http** et **https**.
	Enable proxy server	Cochez cette case si vous vous connectez derrière un proxy et renseignez les informations correspondantes. Disponible pour les protocoles **http**, **https** et **ftp**.
	Enable NTLM Credentials	Cochez cette case si vous utiliser un protocole d'authentification NTLM. **Domain** : Nom de domaine du client. **Host** : Adresse IP du client.

		Disponible pour les protocoles **http**, **https** et **ftp**.
	Need authentication	Cochez la case d'authentification et renseignez le nom de l'utilisateur dans le champs **Username** et son mot de passe dans le champs **Password**, si cela est nécessaire pour accéder au protocole. Disponible pour les protocoles **http** et **https**.
	Support redirection	Cochez cette case pour répéter la requête de redirection jusqu'à ce que la redirection soit effective et que le fichier puisse être récupéré. Disponible pour les protocoles **http**, **https** et **ftp**.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **INPUT_STREAM** : le contenu du fichier récupéré. Cette variable est une variable **Flow** et retourne une classe InputStream. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en composant de départ pour alimenter le flux d'entrée d'un Job et est souvent utilisé avec un lien de type **OnSubjobOk** ou **OnComponentOk**, en fonction du contexte.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario 1 : Récupérer des données à partir d'un protocole HTTP

Ce scénario décrit un Job comprenant trois composants, qui récupère un fichier d'un site internet HTTP, lit des données du fichier récupéré et affiche les données dans la console.

Déposer et relier les composants

1. De la **Palette**, déposez les composants **tFileFetch**, **tFileInputDelimited** et **tLogRow** dans l'espace de modélisation graphique.
2. Reliez le **tFileFetch** au **tFileInputDelimited** à l'aide d'un lien **Trigger** > **OnSubjobOk** ou **OnComponentOk**.
3. Reliez le **tFileInputDelimited** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFileFetch** pour ouvrir sa vue **Basic settings**.

2. Sélectionnez le protocole que vous souhaitez utiliser. Ici, utilisez le protocole **http**.
3. Dans le champ **URI**, saisissez l'URI où le fichier peut être récupéré. Vous pouvez coller l'URI directement dans votre navigateur afin de visualiser les données du fichier.
4. Dans le champ **Destination directory**, sélectionnez le répertoire dans lequel le fichier récupéré doit être stocké. Dans cet exemple, saisissez *D:/Output*.

5. Dans le champ **Destination filename**, saisissez le nouveau nom du fichier si vous souhaitez le renommer. Dans cet exemple, *new.txt*.

6. Si nécessaire, cochez la case **Add header** et définissez une ou plusieurs requêtes HTTP comme conditions de récupération. Par exemple, vous pouvez récupérer le fichier s'il a été modifié depuis une date précise.

   Dans le tableau **Headers**, dans le champ **Name**, saisissez *"If-Modified-Since"* puis saisissez *"Sat, 29 Oct 1994 19:43:31 GMT"* dans le champ **Value**. Pour plus d'informations concernant la définition d'en-têtes HTTP, consultez Header Field Definitions (en anglais).

7. Double-cliquez sur le **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.

8. Dans le champ **File name**, saisissez le chemin d'accès complet au fichier récupéré stocké localement.

9. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir la boîte de dialogue **[Schema]**. Dans cet exemple, ajoutez une colonne *output* pour stocker les données du fichier récupéré.

10. Laissez les autres paramètres tels qu'ils sont.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour l'exécuter.

```
Starting job test at 17:14 10/01/2014.

[statistics] connecting to socket on port 3656
[statistics] connected
Well done
[statistics] disconnected
Job test ended at 17:14 10/01/2014. [exit code=0]
```

Les données du fichier récupéré s'affichent dans la console.

Scénario 2 : Réutiliser un cookie stocké pour récupérer des fichiers via un protocole HTTP

Ce scénario décrit un Job comprenant deux composants, qui se connecte à un site Web HTTP donné et, à l'aide d'un cookie stocké dans un répertoire local spécifié, récupère les données de ce site.

Déposer et relier les composants

1. Déposez deux composants **tFileFetch** de la **Palette** dans l'espace de modélisation graphique.
2. Reliez ces deux composants (en tant que sous-jobs) à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Configurer les composants

Configurer le premier sous-job

1. Double-cliquez sur le **tFileFetch_1** afin d'ouvrir sa vue **Component** et définir ses propriétés.

2. Dans la liste **Procotol**, sélectionnez le protocole que vous souhaitez utiliser. Ici, utilisez le protocole **https**.

3. Dans le champ **URI**, saisissez l'URI via laquelle vous pouvez vous connecter au site Web et récupérer la page Web. Dans cet exemple, l'URI est `https://www.codeproject.com/script/Membership/LogOn.aspx?download=true`.

4. Dans le champ **Destination directory**, parcourez votre répertoire jusqu'au dossier dans lequel stocker la page Web récupérée. Ce dossier sera créé à la volée s'il n'existe pas. Dans cet exemple, saisissez *D:/download*.

5. Dans le champ **Destination filename**, saisissez un nouveau nom pour la page Web si vous souhaitez le modifier. Dans cet exemple, saisissez *codeproject.html*.

6. Dans le tableau **Parameters**, cliquez deux fois sur le bouton **[+]** pour ajouter deux lignes et renseignez les informations d'authentification pour accéder au site Web souhaité.

   Dans la colonne **Name** du tableau **Parameters**, saisissez respectivement le nom des deux lignes, *Email* et *Password*, requises par le site Web auquel vous vous connectez.

   Dans la colonne **Value**, saisissez les informations correspondantes, respectivement votre adresse e-mail et votre mot de passe.

7. Cochez la case **Save cookie**.

8. Dans le champ **Cookie file**, saisissez le chemin d'accès complet au fichier dans lequel vous souhaitez stocker le cookie. Dans cet exemple, le chemin est *D:/download/cookie*.

9. Cliquez sur l'onglet **Advanced settings** pour ouvrir cette vue.

10. Cochez la case **Support redirection** afin que la requête de redirection soit répétée jusqu'à ce que la redirection réussisse.

Configurer le second sous-job

1. Double-cliquez sur le **tFileFetch_2** afin d'ouvrir sa vue **Component** et définir ses propriétés.

2. Dans la liste **Procotol**, sélectionnez **http**.

3. Dans le champ **URI**, saisissez l'adresse à partir de laquelle récupérer les fichiers qui vous intéressent. Dans cet exemple, l'adresse est `http://www.codeproject.com/script/articles/download.aspx?file=/KB/DLL/File_List_Downloader/FLD02June2011_Source.zip&rp=http://www.codeproject.com/Articles/203991/File-List-Downloader`.

4. Dans le champ **Destination directory**, parcourez votre répertoire jusqu'au dossier dans lequel vous souhaitez stocker les fichiers récupérés. Ce dossier peut être automatiquement créé s'il n'existe pas durant le processus d'exécution. Dans cet exemple, saisissez *D:/download*.

5. Dans le champ **Destination Filename**, saisissez un nouveau nom pour le fichier si vous souhaitez le modifier. Dans cet exemple, saisissez *source.zip*.

6. Décochez la case **POST method** pour désactiver le tableau **Parameters**.

7. Cochez la case **Read cookie**.

8. Dans le champ **Cookie file**, parcourez votre système jusqu'au fichier utilisé pour sauvegarder le cookie. Dans cet exemple, le fichier est *D:/download/cookie*.

Sauvegarder et exécuter le Job

1. Sauvegardez votre Job en appuyant sur les touches **Ctrl+S**.

2. Appuyez sur **F6** pour l'exécuter ou cliquez sur le bouton **Run** dans l'onglet **Run**.

   Allez dans le dossier local *D:/download* pour vérifier le fichier téléchargé.

Scénario associé

Pour un exemple de transfert de données en mode stream, consultez *Scénario 2 : Lire les données d'un fichier distant en mode stream*.

tFileInputJSON

Le composant **tFileInputJSON** appartient à deux familles différentes : Internet et File. Pour plus d'informations, consultez *tFileInputJSON*.

tFTPConnection

Propriétés du tFTPConnection

Famille de composant	Internet/FTP	
Fonction	Le composant **tFTPConnection** ouvre une connexion FTP afin d'effectuer une transaction.	
Objectif	Le **tFTPConnection** permet d'ouvrir une connexion FTP afin d'effectuer des transferts de fichiers en une seule transaction.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Adresse IP du serveur FTP.
	Port	Numéro du port d'écoute du serveur FTP.
	Username et *Password*	Informations d'authentification de l'utilisateur du serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	SFTP Support/ Authentication method	Cochez cette case puis, à partir de la liste déroulante **Authentication method**, sélectionnez l'option adéquate d'authentification au serveur : **Password** : Renseignez le mot de passe. **Public key** : Renseignez la clé privée ou cliquez sur le bouton **[...]** à côté du champ **Private key** pour parcourir vos dossiers jusqu'au fichier adéquat. Si vous choisissez l'option **Public Key** comme méthode d'authentification, assurez-vous que la clé ait été ajoutée à l'agent ou qu'aucune passphrase (phrase secrète) ne soit demandée. Lorsque la case est décochée, sélectionnez votre mode de connexion, **Passive** : vous déterminez le port de connexion à utiliser pour permettre le transfert des données, ou **Active** : le serveur FTP détermine le port de connexion à utiliser pour permettre le transfert des données.
Advanced settings	*Use Socks Proxy*	Cochez cette case si vous souhaitez utiliser un proxy, puis paramétrez les champs **Host**, **Port**, **User** et **Password** du proxy.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier.	

	CURRENT_STATUS : résultats d'exécution du composant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé comme sous-job a un seul composant.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénarios associés

Pour un scénario associé, consultez *Scénario : Copier des fichiers sur un serveur FTP distant*.

Consultez également *Scénario : Itération sur un répertoire distant* du **tFTPFileList**.

tFTPDelete

Propriétés du tFTPDelete

Famille de composant	Internet/FTP	
Fonction	Le composant **tFTPDelete** supprime les fichiers sélectionnés via une connexion FTP.	
Objectif	Le **tFTPDelete** supprime les fichiers d'un serveur FTP distant.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection/ Component List	Cochez la case et cliquez sur le composant de connexion adéquat dans la liste **Component list** pour réutiliser les informations de connexion que vous avez déjà définies. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component List** présente uniquement les composants de connexion du Job du même niveau. Si vous souhaitez utiliser une connexion existant dans un autre niveau, vous pouvez utiliser l'onglet **Dynamic settings**, pour partager la connexion. Dans ce cas, vérifiez que le nom de la connexion est unique et distinct à travers les deux niveaux de Job. Pour plus d'informations concernant les **Dynamic settings**, consultez le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP du serveur FTP.
	Port	Numéro du port d'écoute du serveur FTP.
	Username et *Password*	Informations d'authentification de l'utilisateur du serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Remote directory	Répertoire source où se trouvent les fichiers à supprimer.
	Move to the current directory	Cette option apparaît lorsque la case **Use an existing connection** est cochée. Cochez cette case pour naviguer dans le répertoire spécifié dans le champ **Remote directory**. Le composant FTP suivant relié au **tFTPFileList** prend ce répertoire comme racine du répertoire distant lorsque la même connexion est utilisée.
	SFTP Support/ Authentication method	Cochez cette case puis, à partir de la liste déroulante **Authentication method**, sélectionnez l'option adéquate d'authentification au serveur : **Password** : Renseignez le mot de passe. **Public key** : Renseignez la clé privée ou cliquez sur le bouton **[...]** à côté du champ **Private key** pour parcourir vos dossiers jusqu'au fichier adéquat. Si vous choisissez l'option **Public Key** comme méthode d'authentification, assurez-vous que la clé ait été ajoutée

		à l'agent ou qu'aucune passphrase (phrase secrète) ne soit demandée. Lorsque la case est décochée, sélectionnez votre mode de connexion, **Passive** : vous déterminez le port de connexion à utiliser pour permettre le transfert des données, ou **Active** : le serveur FTP détermine le port de connexion à utiliser pour permettre le transfert des données.
	Use Perl5 Regex Expression as Filemask	Cochez cette case pour utiliser des expressions régulières Perl5 comme filtres dans le champ **Files**. Pour plus d'informations concernant la syntaxe des expressions régulières Perl5, consultez la page Perl5 Regular Expression Syntax (en anglais).
	Files	Noms ou chemins d'accès aux fichiers à transférer. Vous pouvez spécifier plusieurs fichiers dans une ligne en utilisant des caractères de remplacement ou des expressions régulières.
Advanced settings	*Use Socks Proxy*	Cochez cette case si vous souhaitez utiliser un proxy, puis paramétrez les champs **Host**, **Port**, **User** et **Password** du proxy.
	Ignore Failure At Quit (FTP)	Cochez cette case afin d'ignorer les erreurs dues à la fermeture du FTP ou de la bibliothèque.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **CURRENT_STATUS** : résultats d'exécution du composant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez *Scénario : Copier des fichiers sur un serveur FTP distant*.

tFTPFileExist

Propriétés du tFTPFileExist

Famille de composant	Internet/FTP	
Fonction	Le composant **tFTPFileExist** vérifie si un fichier est présent sur le serveur FTP.	
Objectif	Le **tFTPFileExist** permet de vérifier la présence d'un fichier sur le serveur FTP distant.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection/ Component List	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Adresse IP du serveur FTP.
	Port	Numéro du port d'écoute du serveur FTP.
	Username et *Password*	Informations d'authentification de l'utilisateur du serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Remote directory	Répertoire source où se trouvent les fichiers à supprimer.
	File Name	Nom du fichier dont vous voulez vérifier la présence.
	SFTP Support/ Authentication method	Cochez cette case puis, à partir de la liste déroulante **Authentication method**, sélectionnez l'option adéquate d'authentification au serveur : **Password** : Renseignez le mot de passe. **Public key** : Renseignez la clé privée ou cliquez sur le bouton **[...]** à côté du champ **Private key** pour parcourir vos dossiers jusqu'au fichier adéquat. Si vous choisissez l'option Public Key comme méthode d'authentification, assurez-vous que la clé ait été ajoutée à l'agent ou qu'aucune passphrase (phrase secrète) ne soit demandée. Lorsque la case est décochée, sélectionnez votre mode de connexion, **Passive** : vous déterminez le port de connexion à utiliser pour permettre le transfert des données, ou **Active** : le serveur FTP détermine le port de connexion à utiliser pour permettre le transfert des données.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
Advanced settings	*Use Socks Proxy*	Cochez cette case si vous souhaitez utiliser un proxy, puis paramétrez les champs **Host**, **Port**, **User** et **Password** du proxy.

	Ignore Failure At Quit (FTP)	Cochez cette case afin d'ignorer les erreurs dues à la fermeture du FTP ou de la bibliothèque.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **EXISTS** : résultat spécifiant si un fichier existe ou non. Cette variable est une variable **Flow** et retourne un booléen. **FILENAME** : nom du fichier traité. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénarios associés

Pour un scénario associé, consultez *Scénario : Copier des fichiers sur un serveur FTP distant*.

Pour un scénario associé utilisant un protocole différent, consultez *Gérer un fichier en utilisant SCP*.

tFTPFileList

Propriétés du tFTPFileList

Famille de composant	Internet/FTP	
Fonction	Le composant **tFTPFileList** fait une boucle sur les fichiers et/ou les dossiers d'un répertoire distant donné.	
Objectif	Le **tFTPFileList** récupère un groupe de fichiers et/ou de dossiers suivant un masque donné et fait une boucle sur chacun d'eux, en se connectant à un répertoire distant via un protocole FTP.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection/ Component List	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Adresse IP.
	Port	Numéro du port d'écoute.
	Username et *Password*	Informations d'authentification de l'utilisateur. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Remote directory	Chemin d'accès au répertoire distant.
	Move to the current directory	Cette option apparaît lorsque la case **Use an existing connection** est cochée. Cochez cette case pour naviguer dans le répertoire spécifié dans le champ **Remote directory**. Le composant FTP suivant relié au **tFTPFileList** prend ce répertoire comme racine du répertoire distant lorsque la même connexion est utilisée.
	File detail	Cochez cette case pour voir toutes les informations du fichier, et non pas juste son nom.
	SFTP Support/ Authentication method	Cochez cette case puis, à partir de la liste déroulante **Authentication method**, sélectionnez l'option adéquate d'authentification au serveur : **Password** : Renseignez le mot de passe. **Public key** : Renseignez la clé privée ou cliquez sur le bouton **[...]** à côté du champ **Private key** pour parcourir vos dossiers jusqu'au fichier adéquat. Si vous choisissez l'option **Public Key** comme méthode d'authentification, assurez-vous que la clé ait été ajoutée à l'agent ou qu'aucune passphrase (phrase secrète) ne soit demandée. Lorsque la case est décochée, sélectionnez votre mode de connexion, **Passive** : vous déterminez le port de connexion à utiliser pour permettre le transfert des données, ou **Active** :

		le serveur FTP détermine le port de connexion à utiliser pour permettre le transfert des données.
	Connect Mode	Sélectionnez le mode adéquat de connexion au serveur : **Active** : vous déterminez le port de connexion à utiliser pour permettre le transfert des données. **Passive** : le serveur FTP détermine le port de connexion à utiliser pour permettre le transfert des données.
Advanced settings	*Use Socks Proxy*	Cochez cette case si vous souhaitez utiliser un proxy, puis paramétrez les champs **Host**, **Port**, **User** et **Password** du proxy.
	Ignore Failure At Quit (FTP)	Cochez cette case afin d'ignorer les erreurs dues à la fermeture du FTP ou de la bibliothèque.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **CURRENT_FILE** : nom du fichier courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_FILEPATH** : nom du fichier courant ainsi que son chemin d'accès. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le composant **tFTPFilelist** est généralement utilisé comme composant seul dans un sous-job mais il peut aussi être utilisé avec d'autres composants.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Itération sur un répertoire distant

Le scénario suivant décrit un Job de trois composants qui a pour objectif de se connecter à un serveur FTP, de lister les fichiers d'un répertoire distant donné en fonction d'un masque, de les récupérer et de les envoyer vers un répertoire local défini.

Déposer et relier les composants

1. Cliquez et déposez les composants suivants : **tFTPConnection**, **tFTPFileList** et **tFTPGet** dans l'espace de modélisation.

2. Cliquez-droit sur le composant **tFTPConnection** et connectez-le au composant **tFTPFileList** via une connexion de type **OnSubjobOk**. Puis connectez le **tFTPFileList** au **tFTPGet** à l'aide d'une connexion de type **Iterate**.

Configurer les composants

Configurer une connexion au serveur FTP

1. Sélectionnez le **tFTPConnection** et cliquez sur l'onglet **Basic settings** de la vue **Component** pour paramétrer le composant :

2. Dans le champ **Host**, renseignez l'adresse IP du serveur FTP.

3. Dans le champ **Port**, renseignez le numéro du port d'écoute.

4. Dans les champs **Username** et **Password**, renseignez respectivement le login et le mot de passe de connexion au serveur distant.

5. Dans la liste **Connect Mode**, sélectionnez le mode de connexion au serveur FTP que vous souhaitez utiliser.

Configurer une liste de téléchargements FTP

1. Sélectionnez le **tFTPFileList** et cliquez sur l'onglet **Basic settings** de la vue **Component** pour paramétrer le composant :

2. Cochez la case **Use an existing connection** et dans la liste **Component list**, sélectionnez le composant de connexion **tFTPConnection_1**. Les informations de connexion sont automatiquement renseignées.

3. Dans le champ **Remote directory**, renseignez le chemin relatif du répertoire contenant les fichiers à lister. Décochez la case **Move to the current directory**.

4. Dans le champ **Filemask**, indiquez un masque de fichier. Utilisez les caractères spéciaux si besoin est. Dans cet exemple, on ne veut récupérer que les fichiers délimités (*.csv*).

Configurer le téléchargement du fichier

1. Sélectionnez le **tFTPGet** et cliquez sur l'onglet **Basic settings** de la vue **Component** pour paramétrer le composant :

2. Cochez la case **Use an existing connection** et dans la liste **Component list**, sélectionnez le composant de connexion adéquat, **tFTPConnection_1** dans cet exemple. Les informations de connexion sont automatiquement renseignées.

3. Dans le champ **Local directory**, renseignez le chemin relatif du répertoire de destination des fichiers à récupérer.

4. Dans le champ **Remote directory**, renseignez le chemin relatif du répertoire contenant les fichiers à récupérer. Décochez la case **Move to the current directory**.

5. Dans la liste **Transfer Mode**, sélectionnez le mode de transfert FTP que vous souhaitez utiliser.

6. Dans la liste **Overwrite file**, sélectionnez l'une des options disponibles pour le transfert du fichier.

7. Dans le champ **Filemask**, sélectionnez la variable globale `((String)globalMap.get("tFTPFileList_1_CURRENT_FILEPATH"))` de manière à traiter tous les fichiers du répertoire source.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** pour l'exécuter.

Une fois la connexion au serveur FTP établie, les fichiers sont listés dans le répertoire spécifié, selon le masque choisi, puis sont récupérés et stockés dans le répertoire de destination préalablement défini. Vous pouvez vérifier que la copie des fichiers a été réalisée à partir de votre explorateur de disque.

tFTPFileProperties

Propriétés du tFTPFileProperties

Famille du composant	Internet	
Fonction	Le composant **tFTPFileProperties** permet de créer un flux monoligne affichant les propriétés du fichier traité sur un serveur FTP.	
Objectif	Le **tFTPFileProperties** permet de recueillir les informations concernant les propriétés principales d'un fichier défini sur un serveur FTP.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Use an existing connection/ Component List	Cochez la case et cliquez sur le composant de connexion adéquat dans la liste **Component list** pour réutiliser les informations de connexion que vous avez déjà définies.
	Host	Adresse IP du serveur FTP.
	Port	Numéro du port d'écoute du serveur FTP.
	Username	Identifiant de connexion au serveur FTP.
	Password	Mot de passe de connexion au serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre,

		saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Remote directory	Chemin d'accès du répertoire source où les fichiers sont récupérés.
	File	Chemin d'accès et nom du fichier à traiter, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	SFTP Support/ Authentication method	Cochez cette case puis, à partir de la liste déroulante **Authentication method**, sélectionnez l'option adéquate d'authentification au serveur : **Password** : Renseignez le mot de passe. **Public key** : Renseignez la clé privée ou cliquez sur le bouton **[...]** à côté du champ **Private key** pour parcourir vos dossiers jusqu'au fichier adéquat. Si vous choisissez l'option Public Key comme méthode d'authentification, assurez-vous que la clé ait été ajoutée à l'agent ou qu'aucune passphrase (phrase secrète) ne soit demandée. Lorsque la case est décochée, sélectionnez votre mode de connexion, **Passive** : vous déterminez le port de connexion à utiliser pour permettre le transfert des données, ou **Active** : le serveur FTP détermine le port de connexion à utiliser pour permettre le transfert des données.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	Calculate MD5 Hash	Cochez cette case pour vérifier le MD5 du fichier téléchargé.
Advanced settings	*Use Socks Proxy*	Cochez cette case si vous souhaitez utiliser un proxy, puis paramétrez les champs **Host**, **Port**, **User** et **Password** du proxy.
	Ignore Failure At Quit (FTP)	Cochez cette case afin d'ignorer les erreurs dues à la fermeture du FTP ou de la bibliothèque.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external	

	+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *Scénario : Afficher les propriétés d'un fichier traité*.

tFTPGet

Propriétés du tFTPGet

Famille de composant	Internet/FTP	
Fonction	Le composant **tFTPGet** récupère les fichiers sélectionnés via une connexion FTP.	
Objectif	Le **tFTPGet** récupère les fichiers sélectionnés d'un serveur FTP distant et les copie dans un répertoire local.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection/ Component List	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Adresse IP du serveur FTP.
	Port	Numéro du port d'écoute du serveur FTP.
	Username	Identifiant de connexion au serveur FTP.
	Password	Mot de passe de connexion au serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Local directory	Chemin d'accès à l'emplacement de destination du fichier.
	Remote directory	Chemin d'accès du répertoire source où les fichiers sont récupérés.
	Move to the current directory	Cette option apparaît lorsque la case **Use an existing connection** est cochée. Cochez cette case pour naviguer dans le répertoire spécifié dans le champ **Remote directory**. Le composant FTP suivant relié au **tFTPFileList** prend ce répertoire comme racine du répertoire distant lorsque la même connexion est utilisée.
	Transfer mode	Différents modes de transfert FTP.
	Overwrite file	Liste des options disponibles pour le transfert du fichier. **Append** : Cochez cette case pour écrire les données à la fin de l'enregistrement, et ainsi ne pas l'écraser.
	SFTP Support	Lorsque vous cochez cette case, les champs **Overwrite file** et **Authentication method** s'affichent. **Overwrite file** vous propose trois options : - **Overwrite** : pour écraser le fichier déjà existant. - **Resume** : pour reprendre le téléchargement du fichier là où il a été interrompu. - **Append** : pour écrire à la suite du fichier sans l'écraser. **Authentication** vous propose deux modes d'authentification :

		- **Public key** : Renseignez le chemin d'accès à la clé publique. - **Password** : Renseignez le mot de passe.
	FTPS Support	Cochez cette case si vous souhaitez vous connecter au serveur FTP via une connexion FTPS. Deux champs apparaissent : **Keystore File** : Saisissez le chemin d'accès au keystore (fichier protégé par un mot de passe, qui peut contenir plusieurs clés et certificats). **Keystore Password** : saisissez votre mot de passe correspondant au keystore.
	Use Perl5 Regex Expression as Filemask	Cochez cette case pour utiliser des expressions régulières Perl5 comme filtres dans le champ **Files**. Pour plus d'informations concernant la syntaxe des expressions régulières Perl5, consultez la page Perl5 Regular Expression Syntax (en anglais).
	Files	Noms ou chemins d'accès aux fichiers à transférer. Vous pouvez spécifier plusieurs fichiers dans une ligne en utilisant des caractères de remplacement ou des expressions régulières.
	Connection Mode	Sélectionnez votre mode de connexion. **Passive** : vous déterminez le port de connexion à utiliser pour permettre le transfert des données. **Active** : le serveur FTP détermine le port de connexion à utiliser pour permettre le transfert des données.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et les lignes contenant les erreurs seront ignorées.
Advanced settings	*Use Socks Proxy*	Cochez cette case si vous souhaitez utiliser un proxy, puis paramétrez les champs **Host**, **Port**, **User** et **Password** du proxy.
	Ignore Failure At Quit (FTP)	Cochez cette case afin d'ignorer les erreurs dues à la fermeture du FTP ou de la bibliothèque.
	Print message	Cochez cette case pour afficher dans la console la liste des fichiers téléchargés.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **CURRENT_STATUS** : résultats d'exécution du composant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **TRANSFER_MESSAGES** : informations transférées du fichier. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé comme sous-Job a un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *Scénario : Copier des fichiers sur un serveur FTP distant*.

Consultez également *Scénario : Itération sur un répertoire distant* du **tFTPFileList**.

tFTPPut

Propriétés du tFTPPut

Famille de composant	Internet/FTP	
Fonction	Le composant **tFTPPut** copie les fichiers sélectionnés via une connexion FTP.	
Objectif	Le **tFTPPut** copie les fichiers sélectionnés d'un répertoire local défini vers un répertoire FTP distant.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection/ Component List	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Adresse IP du serveur FTP.
	Port	Numéro du port d'écoute du serveur FTP.
	Username	Identifiant de connexion au serveur FTP.
	Password	Mot de passe de connexion au serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Local directory	Chemin d'accès au répertoire source du ou des fichier(s).
	Remote directory	Chemin d'accès au répertoire de destination du ou des fichier(s).
	Move to the current directory	Cette option apparaît lorsque la case **Use an existing connection** est cochée. Cochez cette case pour naviguer dans le répertoire spécifié dans le champ **Remote directory**. Le composant FTP suivant relié au **tFTPFileList** prend ce répertoire comme racine du répertoire distant lorsque la même connexion est utilisée.
	Transfer mode	Différents modes de transfert FTP.
	Overwrite file ou *Append*	Liste des options disponibles pour le fichier transféré. **Append** : Cochez cette case pour écrire les données à la fin de l'enregistrement, et ainsi ne pas l'écraser.
	SFTP Support et *Authentication method*	Cochez cette case puis, à partir de la liste déroulante **Authentication method**, sélectionnez l'option adéquate d'authentification au serveur : **Password** : Renseignez le mot de passe. **Public key** : Renseignez la clé privée ou cliquez sur le bouton **[...]** à côté du champ **Private key** pour parcourir vos dossiers jusqu'au fichier adéquat. Si vous choisissez l'option **Public Key** comme méthode d'authentification, assurez-vous que la clé ait été ajoutée

		à l'agent ou qu'aucune passphrase (phrase secrète) ne soit demandée.
	FTPS Support	Cochez cette case pour vous connecter au serveur FTP via une connexion FTPS. Une fois cochée, vous devez renseigner les champs **Keystore File** et **Keystore Password**.
	Keystore File	Spécifiez le chemin d'accès au fichier contenant les données du Keystore. Ce champ apparaît uniquement lorsque la case **FTPS Support** est cochée.
	Keystore Password	Spécifiez le mot de passe du Keystore. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Ce champ apparaît uniquement lorsque la case **FTPS Support** est cochée.
	Use Perl5 Regex Expression as Filemask	Cochez cette case pour utiliser des expressions régulières Perl5 comme filtres dans le champ **Files**. Pour plus d'informations concernant la syntaxe des expressions régulières Perl5, consultez la page Perl5 Regular Expression Syntax (en anglais).
	Files	Cliquez sur le bouton **[+]** pour ajouter une ligne puis remplissez les colonnes. **Filemask**: noms et chemins d'accès aux fichiers à transférer. **New name**: nom à donner au fichier FTP après le transfert.
	Connection Mode	Sélectionnez votre mode de connexion. **Passive** : vous déterminez le port de connexion à utiliser pour permettre le transfert des données. **Active** : le serveur FTP détermine le port de connexion à utiliser pour permettre le transfert des données.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et les lignes contenant les erreurs seront ignorées.
Advanced settings	*Use Socks Proxy*	Cochez cette case si vous souhaitez utiliser un proxy, puis paramétrez les champs **Host**, **Port**, **User** et **Password** du proxy.
	Ignore Failure At Quit (FTP)	Cochez cette case afin d'ignorer les erreurs dues à la fermeture du FTP ou de la bibliothèque.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **CURRENT_STATUS** : résultats d'exécution du composant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_FILE_EXISTS** : existence ou non du fichier courant. Cette variable est une variable **Flow** et retourne un booléen. **TRANSFER_MESSAGES** : informations transférées du fichier. Cette variable est une variable **After** et retourne une chaîne de caractères.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé comme sous-Job a un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Copier des fichiers sur un serveur FTP distant

Ce scénario crée un Job à deux composants permettant d'ouvrir une connexion à un serveur distant afin de copier les fichiers souhaités en une seule transaction.

Déposer et relier les composants

1. Cliquez et déposez les composants **tFTPConnection** permettant d'effectuer toutes les opérations en une seule transaction et **tFTPPut** de la famille **Internet** dans l'espace de modélisation graphique.

2. Reliez le **tFTPConnection** au **tFTPPut** à l'aide d'un lien de type **OnSubjobOk**.

Configurer les composants

Configurer une connexion au serveur FTP

1. Double-cliquez sur le **tFTPConnection** pour paramétrer ses propriétés de base dans l'onglet **Basic settings** de la vue **Component** :

2. Dans le champ **Host**, renseignez l'adresse IP du serveur.

3. Dans le champ **Port**, renseignez le numéro du port d'écoute.

4. Dans les champs **Username** et **Password**, renseignez respectivement le login et le mot de passe de connexion au serveur distant.

5. Dans la liste **Connection Mode**, sélectionnez le mode de connexion au serveur FTP que vous souhaitez utiliser, **Active** dans cet exemple.

Configurer le chargement du fichier sur le serveur FTP

1. Dans l'espace de modélisation graphique, double-cliquez sur le **tFTPPut** pour paramétrer ses propriétés de base dans l'onglet **Basic settings** de la vue **Component** :

2. Cochez la case **Use an existing connection** et dans la liste **Component list**, sélectionnez le composant de connexion **tFTPConnection_1**. Les informations de connexion sont automatiquement renseignées.

3. Dans le champ **Local directory**, renseignez le chemin d'accès au répertoire local contenant le fichier, à moins que vous ne renseigniez directement les masques des fichiers dans la colonne **Filemask** du champ **Files**.

4. Dans le champ **Remote directory**, renseignez les informations de connexion au répertoire du serveur distant. Décochez la case **Move to the current directory**.

5. Dans la liste **Transfert mode**, sélectionnez le mode de transfert utilisé.

6. Dans la liste **Overwrite file**, sélectionnez l'option d'écrasement du fichier.

7. Dans le tableau **Files**, cliquez deux fois sur le bouton **[+]** pour ajouter deux lignes à la liste **Filemask**, renseignez les masques de fichier pour chaque fichier que vous souhaitez copier dans le répertoire distant.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** pour l'exécuter.

Les fichiers spécifiés dans le champ **Filemask** sont copiés dans le répertoire du serveur distant.

tFTPRename

Propriétés du tFTPRename

Famille de composant	Internet/FTP	
Fonction	Le composant **tFTPRename** renomme les fichiers sélectionnés via une connexion FTP.	
Objectif	Le **tFTPRename** renomme des fichiers sur un serveur.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection/ Component List	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Adresse IP du serveur FTP.
	Port	Numéro du port d'écoute du serveur FTP.
	Username	Identifiant de connexion au serveur FTP.
	Password	Mot de passe de connexion au serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Remote directory	Chemin d'accès au répertoire de destination du ou des fichier(s).
	Move to the current directory	Cette option apparaît lorsque la case **Use an existing connection** est cochée. Cochez cette case pour naviguer dans le répertoire spécifié dans le champ **Remote directory**. Le composant FTP suivant relié au **tFTPFileList** prend ce répertoire comme racine du répertoire distant lorsque la même connexion est utilisée.
	Overwrite file	Liste des options disponibles pour le fichier transféré. **Append** : Cochez cette case pour écrire les données à la fin de l'enregistrement, et ainsi ne pas l'écraser.
	SFTP Support et *Authentication method*	**SFTP Support** : cochez cette case pour établir une méthode d'authentification. **Authentication method** : sélectionnez la méthode d'authentification à partir de la liste déroulante.
	Files	Cliquez sur le bouton **[+]** pour ajouter une ligne puis remplissez les colonnes. **Filemask** : noms et chemins d'accès aux fichiers à transférer. **New name** : nom à donner au fichier FTP après le transfert.
	Connection Mode	Sélectionnez votre mode de connexion. **Passive** : vous déterminez le port de connexion à utiliser pour permettre le transfert des données.

		Active : le serveur FTP détermine le port de connexion à utiliser pour permettre le transfert des données.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et les lignes contenant les erreurs seront ignorées.
Advanced settings	*Use Socks Proxy*	Cochez cette case si vous souhaitez utiliser un proxy, puis paramétrez les champs **Host**, **Port**, **User** et **Password** du proxy.
	Ignore Failure At Quit (FTP)	Cochez cette case afin d'ignorer les erreurs dues à la fermeture du FTP ou de la bibliothèque.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **CURRENT_STATUS** : résultats d'exécution du composant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme sous-Job a un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Renommer un fichier situé sur un serveur FTP

Ce scénario permet de renommer un fichier sur un serveur FTP à l'aide du composant **tFTPRename**. Avant d'effectuer le renommage sur le serveur FTP, le fichier se présente comme suit :

Relier les composants

1. Déposez un composant **tFTPConnection** et un **tFTPRename** de la **Palette** dans l'espace de modélisation graphique.

2. Reliez le **tFTPConnection** au **tFTPRename** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Configurer les composants

1. Double-cliquez sur le **tFTPConnection** pour ouvrir sa vue **Basic settings**.

2. Dans les champs **Host** et **Port**, saisissez respectivement l'adresse IP de l'hôte et le numéro du port.

3. Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant et votre mot de passe.

4. Double-cliquez sur le **tFTPRename** pour ouvrir sa vue **Basic settings**.

5. Cochez la case **Use an existing connection** et sélectionnez dans la liste **Component list** le composant de connexion que vous souhaitez utiliser.
6. Dans le champ **Remote directory**, saisissez le chemin d'accès au répertoire contenant le fichier sur le serveur FTP.
7. Dans la table **Files**, cliquez sur le bouton **[+]** pour ajouter une ligne.
8. Dans la colonne **Filemask**, saisissez le nom du fichier.
9. Dans la colonne **New name**, saisissez le nom que vous souhaitez donner au fichier.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** pour l'exécuter.
3. Dans l'interface du client FTP, vous pouvez constater que le nom du fichier a bien été modifié. Le fichier se nomme à présent *action_movies.json*.

Scénario associé

Pour un scénario associé, consultez *Scénario : Copier des fichiers sur un serveur FTP distant*.

tFTPTruncate

Propriétés du tFTPTruncate

Famille de composant	Internet/FTP	
Fonction	Le composant **tFTPTruncate** tronque les fichiers sélectionnés via une connexion FTP.	
Objectif	Le **tFTPTruncate** tronque les fichiers sélectionnés d'un répertoire local défini vers un répertoire FTP distant.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use an existing connection/ Component List	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Adresse IP du serveur FTP.
	Port	Numéro du port d'écoute du serveur FTP.
	Username et *Password*	Informations d'authentification de l'utilisateur au serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Remote directory	Chemin d'accès au répertoire de destination du ou des fichier(s).
	Move to the current directory	Cette option apparaît lorsque la case **Use an existing connection** est cochée. Cochez cette case pour naviguer dans le répertoire spécifié dans le champ **Remote directory**. Le composant FTP suivant relié au **tFTPFileList** prend ce répertoire comme racine du répertoire distant lorsque la même connexion est utilisée.
	SFTP Support et *Authentication method*	**SFTP Support** : cochez cette case pour établir une méthode d'authentification. **Authentication method** : sélectionnez la méthode d'authentification à partir de la liste déroulante.
	Use Perl5 Regex Expression as Filemask	Cochez cette case pour utiliser des expressions régulières Perl5 comme filtres dans le champ **Files**. Pour plus d'informations concernant la syntaxe des expressions régulières Perl5, consultez la page Perl5 Regular Expression Syntax (en anglais).
	Files	Cliquez sur le bouton **[+]** pour ajouter une ligne puis remplissez les colonnes. **Filemask** : saisissez le nom du fichier ou le masque du fichier à l'aide de caractères de remplacement (*) ou d'expressions régulières.
	Connection Mode	Sélectionnez votre mode de connexion. **Passive** : vous déterminez le port de connexion à utiliser pour permettre le transfert des données.

		Active : le serveur FTP détermine le port de connexion à utiliser pour permettre le transfert des données.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
Advanced settings	*Use Socks Proxy*	Cochez cette case si vous souhaitez utiliser un proxy, puis paramétrez les champs **Host**, **Port**, **User** et **Password** du proxy.
	Ignore Failure At Quit (FTP)	Cochez cette case afin d'ignorer les erreurs dues à la fermeture du FTP ou de la bibliothèque.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **CURRENT_STATUS** : résultats d'exécution du composant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme sous-Job a un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Copier des fichiers sur un serveur FTP distant* .

tHttpRequest

Propriétés du tHttpRequest

Famille de composant	Internet	
Fonction	Ce composant envoie une requête HTTP au serveur et obtient des réponses HTTP du serveur.	
Objectif	Le composant **tHttpRequest** vous permet d'envoyer des requêtes HTTP au serveur et d'écrire les réponses localement.	
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur duStudio Talend*.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Sync columns	Cliquez sur ce bouton pour récupérer le schéma du composant précédent.
	URI	Saisissez l'URI (Identifiant uniforme de ressource) à partir duquel seront identifiées les ressources sur le serveur. Un URI est similaire à un URL mais il est plus générique.
	Method	Sélectionnez dans la liste la méthode pour définir l'action à effectuer : **Post** : Choisissez cette méthode pour envoyer des données côté serveur (des données de formulaires HTML par exemple). **Get** : Choisissez cette méthode pour récupérer des données côté serveur.
	Post parameters from file	Parcourez votre système ou entrez le chemin vers le fichier utilisé pour passer des paramètres (corps de la requête) à la méthode POST.
	Write response content to file	Cochez cette case afin de sauvegarder la réponse HTTP dans un fichier local. Vous pouvez soit renseigner le chemin d'accès au fichier d'entrée, soit cliquer sur **[...]** pour choisir le chemin d'accès dans votre système de fichiers.

	Create directory if not exists	Cochez cette case afin de créer le répertoire défini dans le champ **Write response content to file** s'il n'existe pas. Cette case est décochée par défaut et apparaît lorsque la case **Write response content to file** est cochée.
	Headers	Renseignez la (les) paire(s) nom-valeur des en-têtes HTTP pour définir les paramètres de l'opération HTTP requise : **Header key** : Saisissez le nom du champ d'en-tête de l'en-tête HTTP. **Value** : Saisissez le contenu du champ d'en-tête de l'en-tête HTTP. Pour plus d'informations concernant la définition des en-têtes HTTP, consultez (en anglais) : http://en.wikipedia.org/wiki/List_of_HTTP_headers.
	Need authentication	Cochez cette case afin de saisir un identifiant et un mot de passe dans les champs correspondants si une authentification est nécessaire : **user** : Saisissez l'identifiant de connexion au serveur. **password** : Saisissez le mot de passe de connexion au serveur. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **CONNECTED** : indique si une connexion au serveur est établie. Cette variable est une variable **After** et retourne un booléen. **RESPONSE_CODE** : code de réponse retourné par le serveur distant HTTP. Cette variable est une variable **After** et retourne un nombre entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé pour l'envoi de requêtes HTTP au serveur ainsi que pour l'enregistrement des réponses HTTP obtenues. Il peut être utilisé en standalone.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Envoyer une requête HTTP au serveur et sauvegarder localement les réponses HTTP obtenues

Ce scénario décrit un Job à deux composants qui utilise la méthode **GET** pour récupérer des informations du serveur et écrire les réponses dans un fichier local ainsi que dans la console.

Relier les composants

1. Dans la perspective **Integration** du Studio, créez un Job à partir du nœud **Job Designs** du **Repository**.

   Pour plus d'informations concernant la création de Job, consultez le *Guide utilisateur du Studio Talend*.

2. A partir de la **Palette**, déposez les composants suivants dans l'espace de modélisation graphique : un composant **tHttpRequest** et un **tLogRow**.

3. Connectez le composant **tHttpRequest** au composant **tLogRow** à l'aide d'une connexion de type **Row** > **Main**.

Configurer la requête GET

1. Double-cliquez sur le composant **tHttpRequest** pour ouvrir l'onglet **Basic settings** de sa vue **Component** et paramétrer ses propriétés.

2. Saisissez "*http://192.168.0.63:8081/testHttpRequest/build.xml*" dans le champ **URI**. Notez que l'adresse URI fournie dans ce scénario est destinée essentiellement à la démonstration, ce n'est pas une adresse valide.

3. Sélectionnez **GET** dans la liste déroulante **Method**.

4. Cochez la case **Write response content to file** et renseignez le chemin d'accès dans le champ à droite. Pour ce scénario, saisissez *D:/test.txt*.

5. Cochez la case **Need authentication**. Pour ce scénario, saisissez "*tomcat*" dans les champs **user** et **password**.

Exécuter le Job

Vous pouvez à présent exécuter ce Job.

Le composant **tLogRow** est utilisé pour afficher le résultat de l'exécution de ce Job.

1. Si vous souhaitez personnaliser la manière dont le **tLogRow** affiche les résultats, double-cliquez sur le composant pour ouvrir l'onglet **Basic settings** de sa vue **Component** et sélectionnez **Table (print values in cells of a table)** dans le champ **Mode**.

2. Appuyez sur **F6** pour exécuter le Job.

La réponse du serveur est alors sauvegardée et affichée.

```
Starting job test at 15:31 07/03/2011.

[statistics] connecting to socket on port 4011
[statistics] connected
.---------------.
|   tLogRow_1   |
|=--------------=|
|ResponseContent|
|=--------------=|
|hello world!   |
'---------------'

[statistics] disconnected
Job test ended at 15:31 07/03/2011. [exit
code=0]
```

Scénario 2 : Envoyer une requête POST depuis un fichier local JSON

Dans ce scénario, un Job de quatre composants est utilisé pour lire les paramètres depuis un fichier JSON et les envoyer à un site Web dans une requête POST.

Le fichier JSON utilisé se présente comme suit :

```
{"echo":
      [
  {
   "data":"e=hello"
  }
   ]
}
```

Depuis ce fichier, le **tFileInputJSON** lit le paramètre *e* ainsi que sa valeur *hello*. Le **tHttpRequest** envoie le paramètre et sa valeur vers *http://echo.itcuties.com/*, une URL fournie pour l'exemple par une communauté de programmeurs en ligne, www.itcuties.com.

Notez que le paramètre *e* est requis par *http://echo.itcuties.com/*.

Relier les composants

1. Dans la perspective **Integration** du Studio, créez un Job vierge, nommé *httpRequestPostDemo* par exemple, depuis le nœud **Job Designs** du **Repository**.

   Pour plus d'informations concernant la création d'un Job, consultez le *Guide utilisateur du Studio Talend*.

2. Déposez un **tFileInputJSON**, un **tFileOutputDelimited**, un **tHttpRequest** et un **tLogRow** dans l'espace de modélisation graphique.

3. Reliez le **tFileInputJSON** au **tHttpRequest** à l'aide d'un lien **Trigger** > **On Subjob Ok**.

4. Reliez les autres composants entre eux à l'aide de liens **Row** > **Main**.

Lire le fichier JSON

1. Double-cliquez sur le **tFileInputJSON** afin d'ouvrir sa vue **Component**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir l'éditeur de schéma.

3. Cliquez sur le bouton **[+]** afin d'ajouter une ligne et nommez-la, par exemple, *data*.

4. Cliquez sur **OK** afin de valider ces changements puis acceptez la propagation du schéma dans la boîte de dialogue qui s'affiche.

5. Dans le champ **File name**, parcourez votre système ou saisissez le chemin vers le fichier source JSON contenant les paramètres à envoyer.

6. Dans la table **Mapping**, la colonne *data*, définie précédemment dans le schéma du composant, est automatiquement ajoutée. Dans la colonne **JSONPath query** de cette table, saisissez le chemin JSON, entre guillemets, afin d'extraire les paramètres à envoyer. Dans ce scénario, le chemin est `echo[0].data`.

Ecrire le paramètre dans un fichier plat

1. Double-cliquez sur le **tFileOutputDelimited** afin d'ouvrir sa vue **Component**.

2. Dans le champ **File name**, parcourez votre système ou saisissez le chemin vers le fichier plat dans lequel vous souhaitez écrire les paramètres extraits. S'il n'existe pas, le fichier est créé à l'exécution. Dans cet exemple, le chemin est *C:/tmp/postParamsFile.txt*.

Poster le paramètre

1. Double-cliquez sur le **tHttpRequest** afin d'ouvrir sa vue **Component**.

2. Dans le champ **URI**, saisissez l'adresse du serveur vers lequel le paramètre est envoyé, *http://echo.itcuties.com/* dans ce scénario.

3. Dans la liste **Method**, sélectionnez **POST**.

4. Dans le champ **Post parameters from file**, parcourez votre système ou saisissez le chemin vers le fichier plat contenant les paramètres utilisés. Comme défini précédemment, dans le composant **tFileOutputDelimited**, ce chemin est *C:/tmp/postParamsFile.txt*.

Exécuter le Job

Appuyez sur **F6** pour exécuter ce Job.

Le **tLogRow** est utilisé pour afficher les résultats d'exécution du Job.

Une fois le Job exécuté, la vue **Run** s'ouvre automatiquement, vous permettant de vérifier le résultat de l'exécution.

Les réponses renvoyées par le site qui reçoit les paramètres s'affichent dans la console.

tJMSInput

Propriétés du tJMSInput

Famille de composant	Internet	
Fonction	Ce composant crée une interface entre une application Java et un système Middleware orienté Message.	
Objectif	Le **tJMSInput** permet de mettre en place des interactions entre composants applicatifs dans un cadre faiblement couplé, asynchrone et fiable via un serveur JMS.	
Basic settings	*Module List*	Sélectionnez dans cette liste la bibliothèque à utiliser.
	Context Provider	Saisissez l'URL de contexte, par exemple "com.tibco.tibjms.naming.TibjmsInitialContext Factory". Attention cependant, la syntaxe peut varier selon le serveur JMS utilisé
	Server URL	Saisissez l'URL du serveur en respectant la syntaxe, par exemple "tibjmsnaming://localhost:7222".
	Connection Factory JDNI Name	Saisissez le nom JDNI.
	Use Specified User Identity	Si vous devez vous identifier, cochez la case et saisissez votre nom d'utilisateur et votre mot de passe. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Message Type	Sélectionnez le type de message, **Topic** ou **Queue**.
	Message From	Saisissez la source du message, telle qu'attendue par le serveur. Celle-ci peut être composé du type et du nom de la source, par exemple : queue/A ou topic/testtopic. Notez que ce champ est sensible à la casse.
	Timeout for Next Message (in sec)	Renseignez le délai (en secondes) avant de passer au message suivant.
	Maximum Messages	Saisissez le nombre maximal de messages à traiter.
	Message Selector Expression	Spécifiez votre filtre.
	Processing Mode	Sélectionnez le mode de traitement des messages : **Raw Message** : message brut, **Message Content :** contenu du message.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur **Edit schema** afin de visualiser le schéma.
Advanced settings	*Properties*	Cliquez sur le bouton **[+]** sous le tableau afin d'ajouter des lignes contenant le nom et le mot de passe de l'utilisateur requis à l'authentification.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Il requiert un composant de sortie.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Assurez-vous que le serveur JMS correspondant est lancé. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *Scénario : Mettre un message dans une file d'attente du serveur ActiveMQ et le retirer de cette file*.

tJMSOutput

Propriétés du tJMSOutput

Famille de composant	Internet	
Fonction	Ce composant crée une interface entre une application Java et un système Middleware orienté Message.	
Objectif	Le **tJMSOutput** permet de mettre en place des interactions entre composants applicatifs dans un cadre faiblement couplé, asynchrone et fiable via un serveur JMS.	
Basic settings	*Module List*	Sélectionnez dans cette liste la bibliothèque à utiliser.
	Context Provider	Saisissez l'URL de contexte, par exemple "com.tibco.tibjms.naming.TibjmsInitialContext Factory". Attention cependant, la syntaxe peut varier selon le serveur JMS utilisé
	Server URL	Saisissez l'URL du serveur en respectant la syntaxe, par exemple "tibjmsnaming://localhost:7222".
	Connection Factory JDNI Name	Saisissez le nom JDNI.
	Use Specified User Identity	Si vous devez vous identifier, cochez la case et saisissez votre nom d'utilisateur et votre mot de passe. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Message Type	Sélectionnez le type de message, **Topic** ou **Queue**.
	To	Saisissez la cible du message, telle qu'attendue par le serveur.
	Processing Mode	Sélectionnez le mode de traitement des messages : **Raw Message** : message brut, **Message Content** : contenu du message.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma du composant **tJMSInput** est en lecture seule. Il est composé d'une colonne : **Message**.
Advanced settings	*Delivery Mode*	Sélectionnez un mode de réception dans la liste afin d'assurer la qualité des données lors de la réception : **Not Persistent** : Ce mode autorise les pertes de données durant l'échange. **Persistent** : Ce mode assure l'intégrité du message à la réception.
	Properties	Cliquez sur le bouton **[+]** sous le tableau afin d'ajouter des lignes contenant le nom et le mot de passe de l'utilisateur requis à l'authentification.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé en tant que composant de sortie. Il requiert un composant d'entrée.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Assurez-vous que le serveur JMS correspondant est lancé. Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Mettre un message dans une file d'attente du serveur ActiveMQ et le retirer de cette file

Dans ce scénario, le composant **JMSOutput** envoie un message dans une file du serveur ActiveMQ, qui est ensuite récupéré par le **JMSInput**. Ce message est affiché dans la console, via le **tLogRow**.

Relier les composants

1. De la **Palette**, déposez un **tFixedFlowInput**, un **JMSOutput**, un **JMSInput** et un **tLogRow** dans l'espace de modélisation graphique.
2. Reliez le **tFixedFlowInput** au **JMSOutput** à l'aide d'un lien **Row** > **Main**.
3. Reliez le **tFixedFlowInput** au **JMSInput** à l'aide d'un lien **OnSubjobOk**.
4. Connectez le **JMSInput** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

   Sélectionnez **Use Inline Content (delimited file)** dans la zone **Mode**.

   Dans le champ **Content**, saisissez le contenu du message à envoyer au serveur ActiveMQ :

```
message transferred
```

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur de schéma.

3. Cliquez sur le bouton **[+]** pour ajouter une colonne, que vous nommez *messageContent*, de type String.

   Cliquez sur **OK** pour valider la configuration et fermer l'éditeur.

4. Une boîte de dialogue s'ouvre et vous propose de propager le schéma.

   Cliquez sur **Yes** afin de propager le schéma au composant suivant.

5. Double-cliquez sur le **JMSOutput** pour ouvrir sa vue **Basic settings**.

6. Dans la liste **Module List**, sélectionnez la bibliothèque à utiliser, le Jar *activemq* dans cet exemple.

7. Dans le champ **Context Provider**, saisissez l'URI de contexte, *"org.apache.activemq.jndi.ActiveMQInitialContextFactory"* dans ce scénario.

8. Dans le champ **Server URL**, saisissez l'URI du serveur ActiveMQ.

9. Dans le champ **Connection Factory JDNI Name**, saisissez le nom JDNI, *"QueueConnectionFactory"* dans cet exemple.

10. Cochez la case **Use Specified User Identity** afin d'afficher les champs **User Name** et **Password**, dans lesquels vous pouvez saisir respectivement votre identifiant et votre mot de passe.

11. Dans la liste **Message type**, sélectionnez **Queue**.

12. Dans la liste **Processing Mode**, sélectionnez **Message Content**.

13. Configurez le **JMSInput** de la même manière.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter le Job. Le serveur ActiveMQ a démarré à l'adresse suivante : *tcp://192.168.30.46:61616*.

```
[statistics] connecting to socket on port 3999
[statistics] connected
Ready to receive message
Waiting...
message transferred
```

Dans la console, vous pouvez constater que le message est bien transféré et affiché.

Scénario associé

Pour un scénario associé, consultez *Scénario 1 : Communication asynchrone via un serveur MOM* et *Scénario 2 : Transmettre des fichiers XML via un serveur MOM*.

tMicrosoftMQInput

Propriétés du tMicrosoftMQInput

Famille de composant	Internet/MOM et JMS	
Fonction	Le composant **tMicrosoftMQInput** récupère le premier message d'une file d'attente de message Microsoft donnée (le composant supporte uniquement le type String).	
Objectif	Ce composant vous permet de récupérer les messages un par un dans l'ordre de leur ID, dans la file d'attente de message Microsoft. Chaque exécution récupère un seul message.	
Basic settings	*PROPERTY*	Toujours **Built-in**. Le schéma contient une colonne en lecture seule de type String nommée *message*.
	Host	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Queue	Saisissez le nom de la file d'attente à partir de laquelle vous souhaitez récupérer les messages.
	Queue type	Sélectionnez un type de file d'attente de message Microsoft dans la liste déroulante. • **Private** : une file d'attente privée, qui est enregistrée sur la machine locale. • **Public** : une file d'attente publique, qui est enregistrée dans le service de répertoire. Pour plus d'informations concernant les types de file d'attente de message Microsoft, consultez http://msdn.microsoft.com/en-us/library/ms706878%28v=vs.85%29.aspx.
	Die on error	Cette case est décochée par défaut, pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job. Il doit être lié à un composant de sortie.	
Connexions	Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate. **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Iterate **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error.	

	Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.
Limitation	Ce composant ne supporte que le type String et fonctionne uniquement sous Windows. Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénario : Ecrire et récupérer des messages dans une file d'attente de message Microsoft

Ce scénario se compose de deux Jobs. Le premier Job poste des messages dans une file de message Microsoft et le second Job récupère les messages du serveur.

Poster des messages dans une file de messages Microsoft

Dans le premier Job, un message de type string est créé via un **tRowGenerator** et posté dans une file d'attente de message Microsoft à l'aide d'un composant **tMicrosoftMQOutput**. Un composant **tLogRow** intermédiaire affiche le flux passé d'un composant à l'autre.

Déposer et relier les composants

1. Déposez de la **Palette** dans l'espace de modélisation graphique un **tRowGenerator**, un **tLogRow** et un **tMicrosoftMQOutput**.
2. Connectez les composants à l'aide de liens **Row** > **Main**.
3.

Configurer les composants

1. Double-cliquez sur le **tRowGenerator** afin d'ouvrir son éditeur.

2. Dans cet éditeur, cliquez sur le bouton **[+]** pour ajouter trois lignes au schéma.

3. Dans la colonne **Column**, saisissez un nouveau nom pour chaque ligne, afin de les renommer. Ici, saisissez respectivement *ID*, *Name* et *Address*.

4. Dans la colonne **Type**, sélectionnez **Integer** pour la ligne *ID* dans la liste déroulante et laissez les autres lignes en **String**.

5. Dans la colonne **Functions**, sélectionnez `random` pour la ligne *ID*, `getFirstName` pour la ligne *Name* et `getUsCity` pour la ligne *Address*.

6. Dans le champ **Number of Rows for RowGenerator**, saisissez *12* afin de limiter le nombre de lignes à générer. Puis cliquez sur **OK** pour valider vos modifications.

   Lors d'une utilisation réelle, vous devrez utiliser un composant d'entrée pour charger les données qui vous intéressent, au lieu du **tRowGenerator**.

7. Double-cliquez sur le composant **tMicrosoftMQOutput** afin d'ouvrir sa vue **Component**.

8. Dans le champ **Host**, saisissez l'adresse de l'hôte. Dans cet exemple, saisissez *localhost*.

9. Dans le champ **Queue**, saisissez le nom de la file de messages dans laquelle vous souhaitez écrire le message. Dans cet exemple, nommez-la *AddressQueue*.

10. Dans le champ **Message column (String Type)**, sélectionnez *Address* dans la liste déroulante afin de déterminer le corps du message à écrire.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** pour exécuter le Job.

```
Starting job tMicrosoftMQ at 14:03 18/11/2010.

[statistics] connecting to socket on port 3852
[statistics] connected
Queue open failure: Cannot open queue. (hr=MQ_ERROR_QUEUE_NOT_FOUND)
OpenQueueWithAccess (DIRECT=OS:localhost\private$\AddressQueue)
open: fmtname(DIRECT=OS:localhost\private$\AddressQueue) accessmode(1)
sharemode(0)
..............................................................................
.............................................attempting to create queue with
name= '.\private$\AddressQueue', label='Created by
org.talend.msmq.MsmqUtil.java'
OpenQueueWithAccess (DIRECT=OS:.\private$\AddressQueue)
open: fmtname(DIRECT=OS:.\private$\AddressQueue) accessmode(1) sharemode(0)
open: fmtname(DIRECT=OS:.\private$\AddressQueue) accessmode(2) sharemode(0)
7|Theodore|Atlanta
8|Andrew|Saint Paul
7|Benjamin|Oklahoma City
3|Abraham|Providence
4|Rutherford|Lincoln
6|Chester|Little Rock
11|Richard|Lincoln
8|Bill|Boston
9|Herbert|Jackson
10|James|Honolulu
5|Jimmy|Montpelier
9|Woodrow|Nashville
[statistics] disconnected
Job tMicrosoftMQ ended at 14:03 18/11/2010. [exit code=0]
```

Vous pouvez voir que la file a été créée automatiquement et que les messages ont été écrits.

Récupérer le premier message de la file de messages

Créez ensuite le second Job pour récupérer le premier message de la file d'attente.

Déposer et relier les composants

1. Déposez un **tMicrosoftMQInput** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Reliez ces deux composants à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tMicrosoftMQInput** afin d'ouvrir sa vue **Component**.

2. Dans le champ **Host**, saisissez le nom ou l'adresse de l'hôte. Ici, saisissez *localhost*.

3. Dans le champ **Queue**, saisissez le nom de la file d'attente de message à partir de laquelle vous souhaitez récupérer le message. Dans cet exemple, saisissez *AddressQueue*.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour exécuter le Job.

```
Starting job MQInput at 16:57 18/11/2010.

[statistics] connecting to socket on port 3719
[statistics] connected
OpenQueueWithAccess
(DIRECT=OS:localhost\private$\AddressQueue)
open:
fmtname(DIRECT=OS:localhost\private$\AddressQueue)
accessmode(1) sharemode(0)
open:
fmtname(DIRECT=OS:localhost\private$\AddressQueue)
accessmode(2) sharemode(0)
Atlanta
[statistics] disconnected
Job MQInput ended at 16:57 18/11/2010. [exit code=0]
```

Le corps du message *Atlanta* récupéré de la file d'attente s'affiche dans la console.

tMicrosoftMQOutput

Propriétés du tMicrosoftMQOutput

Famille de composant	Internet/MOM et JMS	
Fonction	Le composant **tMicrosoftMQOutput** écrit une colonne définie d'un flux d'entrée de données dans une file de messages Microsoft (le composant supporte uniquement le type String).	
Objectif	Ce composant permet d'écrire des messages dans une file d'attente de messages Microsoft.	
Basic settings	*PROPERTY*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Host	Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.
	Queue	Saisissez le nom de la file de messages dans laquelle vous souhaitez écrire un message. Cette file peut être créée automatiquement à la volée si elle n'existe pas.
	Queue type	Sélectionnez un type de file d'attente de message Microsoft dans la liste déroulante. • **Private Non-Transaction** : une queue privée non-transactionnelle. • **Private Transaction** : une queue privée transactionnelle. • **Public Non-Transaction** : une queue publique non-transactionnelle. • **Public Transaction** : une queue publique transactionnelle. Pour plus d'informations concernant les type de file d'attente de message Microsoft, consultez http://msdn.microsoft.com/en-us/library/ms706878%28v=vs.85%29.aspx.
	Message column	Sélectionnez le message à écrire dans la file de messages Microsoft. Le message sélectionné doit être de type String.
	Die on error	Cette case est décochée par défaut, pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être lié à un composant d'entrée ou intermédiaire.	
Connexions	Liens de sortie (de ce composant à un autre) :	

	Row : Main, Iterate. **Trigger** : Run if, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.
Limitation	Le message à écrire ne peut être null. Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un scénario associé, consultez *Scénario : Ecrire et récupérer des messages dans une file d'attente de message Microsoft*.

tMomCommit

Propriétés du tMomCommit

Famille de composant	Internet	
Fonction	Le composant **tMomCommit** commite des données sur le serveur MQ.	
Objectif	En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.	
Basic settings	*Component List*	Sélectionnez dans la liste le composant de connexion du le Job en cours.
	Close connection	Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. *Si vous utilisez un lien de type **Row** > **Main** pour relier le **tMomCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.*
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Mom, notamment avec le **tMomConnection** et le **tMomRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour plus d'informations relatives au fonctionnement du composant **tMomCommit**, consultez *Scénario 1 : Insérer des données dans des tables mère/fille*

tMomConnection

Propriétés du tMomConnection

Famille de composant	Internet	
Fonction	Le composant **tMomConnection** ouvre une connexion à un serveur MQ.	
Objectif	Ce composant ouvre une connexion vers un serveur MQ dans un but de communication.	
Basic settings	*Failover*	Cochez cette case pour définir les différents serveurs MQ pour le failover et la table d'adresses du serveur affichée. Cette case est disponible si vous sélectionnez **ActiveMQ** dans la liste déroulante **MQ server**.
	MQ server	Sélectionnez dans la liste déroulante le type des serveurs MQ à connecter, **ActiveMQ** ou **WebSphere MQ**.
	Host	Adresse IP du serveur MQ.
	Port	Port d'écoute du serveur MQ.
	Username et *Password*	Informations d'authentification de l'utilisateur. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use Transacted	Cochez cette case pour configurer la session pour laquelle effectuer une transaction. Cette case est disponible si vous sélectionnez **ActiveMQ** dans la liste déroulante **MQ server**.
	Channel	Spécifiez le nom du canal dans lequel la connexion est établie. La valeur par défaut est *DC.SVRCONN*. Ce champ est disponible si vous sélectionnez **WebSphere MQ** dans la liste déroulante du serveur **MQ server**.
	QueueManager	Spécifiez le nom du gestionnaire de file auquel la connexion est établie. Ce champ est disponible si vous sélectionnez **WebSphere MQ** dans la liste déroulante du serveur **MQ server**.
	Use or register a shared Connection	Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant. *Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un **tRunJob** ayant l'une de ces options activée fera échouer votre Job.*
	Shared Connection Name	Saisissez le nom de la connexion partagée. Ce champ est disponible si vous cochez la case **Use or register a shared Connection**.
Advanced settings	*Additional options*	Définissez dans la table les option de transport pour le failover. Cette table est disponible uniquement si vous avez coché la case **Failover** dans la vue **Basic settings**.

		• **Option name** : nom de l'option. • **Option value** : valeur de l'option. Par exemple, saisissez *maxReconnectAttempts* dans le champ **Option name** puis *0* dans le champ **Option value** afin qu'aucune tentative ne soit effectuée pour récupérer la connexion lorsqu'aucun serveur n'est disponible pour le mode failover. Pour plus d'informations concernant les options de transport, consultez http://activemq.apache.org/failover-transport-reference.html (en anglais).
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec des composants Mom, notamment les composants **tMomCommit** et **tMomRollback**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Ce composant supporte les serveurs ActiveMQ et WebSphereMQ.	

Scénario associé

Pour un scénario associé au **tMomConnection**, consultez la *tMysqlConnection*.

tMomInput

Propriétés du tMomInput

Famille de composant	Internet	
Fonction	Ce composant récupère un message d'une file d'attente d'un système Middleware orienté Message et le passe au composant suivant.	
Objectif	Le **tMomInput** permet de mettre en place des communications asynchrones via un serveur MOM.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Keep Listening	Cochez cette case pour consulter en continu le serveur MOM pour récupérer les nouveaux messages. - Si vous sélectionnez le serveur **JBoss Messaging**, cocher cette case fera apparaître le champ **Sleeping time (in sec)**. - Si vous sélectionnez le serveur **ActiveMQ**, cocher cette case fera disparaître le champ **Sleeping time (in sec)**.
	Sleeping time (in sec)	Saisissez en secondes la fréquence de variation. Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.
	MQ Server	Sélectionnez dans cette liste le serveur MOM à utiliser. Les paramètres requis diffèrent légèrement en fonction du serveur sélectionné.
	Host/Port	Renseignez le nom de l'hôte (Host) et l'adresse IP du serveur MOM ainsi que le Port.
	Username	Identifiant de connexion au serveur MQ.
	Password	Mot de passe de connexion au serveur MQ. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Message From	Saisissez la source du message, telle qu'attendue par le serveur. Celle-ci peut être composé du type et du nom de la source, par exemple : queue/A ou topic/testtopic. Notez que ce champ est sensible à la casse. Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.
	Message Type	Sélectionnez le type de message : **topic** ou **queue**. Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.
	Message Body Type	Sélectionnez le type de message : **Text**, **Bytes** ou **Map**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant.

		Le schéma du composant **tMomInput** est composé de deux colonnes : **From** et **Message**, dont le nom est en lecture seule.
Websphere MQ	*Channel*	Renseignez ce champ avec le nom du canal avec lequel la connexion est établie. La valeur par défaut est *DC.SVRCONN*.
	Queue Manager	Programme système fournissant un conteneur logique pour la file de messages et responsable de transférer les données à d'autres gestionnaires de file via des canaux de messages. Renseignez ce champ avec le nom du gestionnaire de file(s) d'attente auquel la connexion est effectuée.
	Message Queue	File d'attente à partir de laquelle les applications de files d'attente de messages peuvent charger des messages et en récupérer. Renseignez ce champ avec le nom de la file de messages.
	Is using message id to fetch	Cochez cette case pour appliquer un filtre portant sur les messages id, filtre que vous déterminerez dans le champ **Message id**.
	Commit (delete message after reading from the queue)	Cochez cette case afin de forcer le commit après lecture de chaque message dans la file d'attente.
	Backout removed messages	Cochez cette case pour indiquer au gestionnaire de file que tous les messages lus depuis le serveur ne seront pas supprimés lorsque la connexion au serveur sera suspendue. Cette case ainsi que la case **Browse message** dans la vue **Advanced settings** vous permettent de lire des messages de manière non destructive dans la file. Elle n'est visible que lorsque la case **Keep listening** est décochée. Pour plus d'informations, consultez le Guide MQSeries Using Java (en anglais).
ActiveMQ	*Receive number of messages*	Cochez cette case afin de limiter le nombre de messages à recevoir dans la console. *Lorsque vous choisissez de limiter le nombre de messages à recevoir, la limite de temps devient inactive, les champs **Keep listening/Sleeping time (in sec)** disparaissent.*
	Failover	Cochez cette case pour afficher la table d'adresses du serveur. Dans cette table, vous pouvez définir plusieurs serveurs MQ pour le failover.
Advanced settings	*Additional options*	Définissez dans la table les options de transport pour le failover. Cette table est disponible uniquement si vous avez coché la case **Failover** dans la vue **Basic settings**. • **Option name** : nom de l'option. • **Option value** : valeur de l'option. Par exemple, saisissez *maxReconnectAttempts* dans le champ **Option name** puis *0* dans le champ **Option value** afin qu'aucune tentative ne soit effectuée pour récupérer la connexion lorsqu'aucun serveur n'est disponible pour le mode failover. Pour plus d'informations concernant les options de transport, consultez http://activemq.apache.org/failover-transport-reference.html (en anglais).
	Get Jms Header	Cochez cette case pour recevoir les en-têtes JMS via le mapping des champs JMS vers les champs MQ Series sur le serveur MQ. Lorsque cette case est cochée, vous pouvez spécifier le nom de l'en-tête JMS ainsi que le nom de la colonne de référence correspondante dans la (les) ligne(s) ajoutée(s) en cliquant sur le bouton **[+]** sous la table **Parameters**. Pour plus d'informations concernant l'utilisation des en-têtes JMS, consultez le Chapitre 12 de MQSeries Using Java (en anglais). Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.

	Get Jms Properties	Cochez cette case pour recevoir les propriétés JMS mappées dans les champs MQMD. Lorsque cette case est cochée, vous pouvez spécifier le nom, le type et le nom de la colonne de référence dans la (les) ligne(s) ajoutée(s) en cliquant sur le bouton **[+]** sous la table **Parameters**. Pour plus d'informations concernant l'utilisation des propriétés JMS, consultez le Chapitre 12 de MQSeries Using Java. Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.
	Browse message	Cochez cette case afin de désactiver les options **Commit (delete message after reading from the queue)** et **Backout removed messages** dans la vue **Basic settings** et d'ouvrir la file pour parcourir les messages. Cette case ainsi que la case **Backout removed messages** dans la vue **Basic settings** vous permettent de lire des messages de manière non destructive dans la file. La case **Browse message** est visible uniquement lorsque la case **Backout removed messages** est décochée. Pour plus d'informations, consultez MQC: MQOO_BROWSE (en anglais).
	Get MQMD Fields	Cochez cette case pour définir un ou plusieurs Message Descriptor(s) en ajoutant de nouveaux champs pour MQMD (Message queuing message descriptor) dans le tableau **Parameter** : **Field Name** : sélectionnez un ou plusieurs descripteur(s) de messages dans la liste afin de récupérer les informations de l'en-tête du message. **Reference Column Name** : Informations d'en-tête et propriétés du message. Pour plus d'informations, consultez Fields for MQMD (en anglais). Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.
	Include Header	Cochez cette case pour définir l'en-tête du message : **MQRFH2 fixed Portion** : Cochez cette case et cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) pour spécifier la **Version**, l'encodage (**Encoding**), le **Format**, les **Flags** et les valeurs **NameValueCCSID** et **CodeCharSetId** pour la portion fixe de l'en-tête MQRFH2. **MCD folder** : Cochez cette case et cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) pour spécifier le nom et la valeur des propriétés décrivant le format du message. **JMS folder** : Cochez cette case et cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) pour spécifier le nom du champ et sa valeur pour le transport des champs d'en-têtes JMS et des propriétés JMSX. **USR folder** : Cochez cette case et cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) pour spécifier le nom du champ et sa valeur pour le transport des propriétés définies par l'application associée au message. Pour plus d'informations concernant les en-têtes MQRFH2, consultez http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r1m0/index.jsp?topic=%2Fcom.ibm.etools.mft.doc%2Faq06920_.htm. Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.

	Set CipherSpec	Cochez cette case pour activer la liste **CipherSpec** dans laquelle vous pouvez spécifier le CipherSpec à utiliser avec le SSL WebSphere. Pour plus d'informations concernant CipherSpec, consultez Specifying CipherSpecs (en anglais). Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion au serveur MQ parmi celles prévues dans votre Job. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Il requiert un composant de sortie.	
Limitation	Assurez-vous que le serveur Websphere, JBoss ou ActiveMQ correspondant est lancé. Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario 1 : Communication asynchrone via un serveur MOM

Ce scénario est composé de deux Jobs. Le premier permet de poster des messages sur la file d'attente du serveur JBoss et le deuxième récupère les messages de ce serveur.

Construire et exécuter le premier Job

Dans le premier Job, un message composé de chaînes de caractères est créé à l'aide du composant **tRowGenerator** puis transmis au serveur JBoss à l'aide du composant **tMomOutput**. Un composant intermédiaire, le composant **tLogRow**, permet d'afficher le flux transmis.

1. Cliquez et déposez ces trois composants dans votre premier Job et connectez-les via le clic-droit avec des liens de type **Main** > **Row**.

2. Double-cliquez sur le composant **tRowGenerator** pour configurer le schéma des données qui seront générées de manière aléatoire.

3. Créez une seule colonne appelée *message*. Elle correspond au message qui sera mis sur la file d'attente du MOM.

4. Cette colonne est de type **String** et peut contenir une valeur nulle. Pour produire les données, utilisez une fonction prédéfinie permettant de concaténer de manière aléatoire les caractères ascii choisis pour former la chaîne de 6 caractères. Cette fonction est la suivante : `getAsciiRandomString`. (version Java). Cliquez sur le bouton d'aperçu pour afficher un échantillon au hasard des données générées.

5. Paramétrez le champ **Number of rows to be generated** permettant de définir le nombre de lignes générée à *10*.

   Cliquez sur **OK** pour valider.

6. Le composant **tLogRow** est uniquement utilisé pour afficher un état intermédiaire des données manipulées. Dans cet exemple, aucune configuration spécifique n'est nécessaire.

7. Sélectionnez le composant **tMomOutput**.

8. Dans cet exemple, le serveur **MQ server** à utiliser est de type **JBoss**.

   Dans les champs **Host** et **Port**, renseignez les informations de connexion correspondantes.

9. Sélectionnez le type de message dans la liste déroulante **Message type**, il peut être de type **Queue** ou **Topic**. Pour cet exemple, sélectionnez le type **Queue** dans la liste.

10. Dans le champ **To**, saisissez les informations source du message strictement telles qu'elles sont attendues par le serveur. Il doit correspondre au type de message que vous avez sélectionné dans la liste déroulante **Message Type**, notamment : `queue/A`.

Le nom du message est sensible à la casse, ainsi `queue/A` et `Queue/A` sont différents.

11. Puis cliquez sur **Sync Columns** pour récupérer le schéma du composant précédent. Le schéma étant en lecture seule, il ne peut être modifié. Les données envoyées sur le serveur MQ proviennent de la première colonne rencontrée du schéma.

12. Exécutez le Job en appuyant sur **F6** et, grâce au composant **tLogRow**, observez dans la console le flux de données transmis.

```
Starting job Mominput at 17:46 14/09/2007.
c8X5GC
lEhC41
opGYqP
x6gOt8
ESknZp
tXU2ET
6H7Nwl
8UA9eM
2sebwV
rfPZAP
Job Mominput ended at 17:46 14/09/2007. [exit code=0]
```

Construire et exécuter le second Job

Puis paramétrez le deuxième Job pour récupérer les messages de la file d'attente du serveur MOM.

1. Cliquez et déposez un composant **tMomInput** (dans la famille **Internet** de la **Palette**) et un composant **tLogRow** pour afficher les messages récupérés.

2. Sélectionnez le **tMomInput** et définissez ses paramètres.

3. Sélectionnez le serveur **MQ server** dans la liste. Dans cet exemple, un serveur de message JBoss est utilisé.

4. Renseignez les informations **Host** et **Port** du serveur.

5. Renseignez les paramètres des champs **Message From** et **Message Type** afin qu'ils correspondent à la source et au type de message attendus par le serveur de message.

6. Le schéma dans le champ **Schema** est en lecture seule et est composé de deux colonnes : **From** et **Message**.

7. Cochez la case **Keep listening** et paramétrez la fréquence de vérification à *5* secondes.

Lorsque vous utilisez l'option **Keep Listening**, vous devez cliquez sur le bouton **Kill** pour stopper le Job.

8. Laissez les paramètres par défaut du composant **tLogRow**.

9. Enregistrez le Job et exécutez-le (si vous le lancez pour la première fois ou si vous l'avez stoppé au cours d'un lancement précédent).

```
Starting job momoutput at 17:47 14/09/2007.
[statistics] connecting to socket on port 3414
[statistics] connected
Ready to receive message
Waiting...
queue/A|c8X5GC
queue/A|1EhC41
queue/A|opGYqP
queue/A|x6gOt8
queue/A|ESknZp
queue/A|tXU2ET
queue/A|6H7Nwl
queue/A|8UA9eM
queue/A|2sebwV
queue/A|rfPZAP
Job momoutput ended at 17:47 14/09/2007. [exit code=(
```

Les messages récupérés sur le serveur sont affichés dans la console.

Scénario 2 : Transmettre des fichiers XML via un serveur MOM

Ce scénario décrit un Job à cinq composants, comprenant deux sous-jobs envoyant des fichiers XML depuis un dossier local vers une file MOM puis récupérant les fichiers de la file MOM et affichant le contenu des fichiers dans la console.

Déposer et relier les composants

1. De la **Palette**, déposez les composants suivants dans l'espace de modélisation graphique : un **tFileList**, un **tFileInputXML**, un **tMomOutput**, un **tMomInput** et un **tLogRow**.

2. Reliez le **tFileList** au **tFileInputXML** à l'aide d'un lien **Row** > **Iterate** et reliez le **tFileInputXML** au **tMomOutput** à l'aide d'un lien **Row** > **Main** afin de former le premier sous-job. Ce sous-job lit chaque fichier XML dans un dossier donné et l'envoie dans une file MOM.

3. Connectez le **tMomInput** au **tLogRow** à l'aide d'un lien **Row** > **Main** pour former le second sous-job. Ce sous-job récupère les fichiers XML de la file MOM et affiche leur contenu dans la console.

4. Reliez le **tFileInputXML** au **tMomInput** à l'aide d'un lien **Trigger** > **On Component Ok** afin de relier les deux sous-jobs.

Configurer le premier sous-job

Configurer les composants d'entrée

1. Double-cliquez sur le **tFileList** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **Directory**, saisissez le chemin d'accès au répertoire duquel lire les fichiers XML, ou parcourez votre système en cliquant sur le bouton **[...]**.

3. Cochez la case **Use Glob Expressions as Filemask**, ajoutez une nouvelle ligne dans la table **Files** en cliquant sur le bouton **[+]** et saisissez `"*.xml"` comme masque de fichier, dans la colonne **Filemask**, afin d'utiliser tous les fichiers XML du répertoire. Laissez les autres paramètres tels qu'ils sont.

4. Double-cliquez sur le composant **tFileInputXML** pour ouvrir sa vue **Basic settings**.

5. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir la boîte de dialogue **[Schema]**.

6. Cliquez sur le bouton **[+]** pour ajouter une colonne, nommez-la *message* dans cet exemple et sélectionnez **Document** dans la liste **Type** afin de gérer les fichiers au format XML. Cliquez sur **OK** pour fermer la boîte de dialogue.

7. Dans le champ **File name/Stream**, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables globales et sélectionnez `tFileList_1.CURRENT_FILEPATH` afin d'effectuer une boucle sur le répertoire des fichiers de contexte.

8. Dans le champ **Loop XPath query**, saisissez `"/"` pour définir la racine comme nœud de boucle dans la structure des fichiers d'entrée. Dans la table **Mapping**, dans la colonne **XPath query**, saisissez `"."` pour extraire toutes les données du nœud de contexte des fichiers source et cochez la case **Get Nodes** afin de construire un flux de données de type **Document**.

Configurer le composant tMomOutput

1. Double-cliquez sur le composant **tMomOutput** pour ouvrir sa vue **Basic settings**.

2. Sélectionnez **WebSphere MQ** dans la liste **MQ server** et saisissez le nom de l'hôte ou l'adresse IP du serveur MQ, ainsi que le numéro du port.

3. Saisissez les informations d'authentification dans les champs **Username** (identifiant) et **Password** (mot de passe) puis saisissez le nom du canal de la file de transmission, dans le champ **Channel**.

4. Comme vous gérez des messages de fichiers, sélectionnez **Text Message** dans la liste **Message Body Type**.

5. Cliquez sur **Sync columns** afin de récupérer la structure du schéma du composant précédent.

6. Renseignez les informations relatives au gestionnaire de file, dans le champ **QueueManager** et les informations concernant la file de messages, dans le champ **MessageQueue**. Laissez les autres paramètres tels qu'ils sont.

Configurer le second sous-job

1. Double-cliquez sur le **tMomInput** pour ouvrir sa vue **Basic settings**.

2. Configurez les paramètres simples du composant de la même manière que pour le **tMomOutput**, y compris les informations du serveur MQ, les informations d'authentification, le canal, le type de message, le gestionnaire de file et la file de messages.

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir la boîte de dialogue **[Schema]**.

4. Dans la liste **Type** de la colonne **message**, sélectionnez **Document** afin de gérer les formats XML puis cliquez sur **OK** pour fermer la boîte de dialogue.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur **Run** dans l'onglet **Run** pour exécuter le Job.

   Les fichiers XML du dossier spécifié sont écrits dans la file de messages puis sont récupérés de la file. Le contenu des fichiers est affiché dans la console.

Execution

Run Kill Clear

```
Ready to receive message
Waiting...
queue|<?xml version="1.0" encoding="UTF-8"?>
<bookstore id="1">
    <bookshelf>
        <category>Languages</category>
        <quantity>200</quantity>
    </bookshelf>
    <bookshelf>
        <category>Arts</category>
        <quantity>300</quantity>
    </bookshelf>
    <bookshelf>
        <category>Science</category>
        <quantity>400</quantity>
    </bookshelf>
</bookstore>
Closing connection
Closing connection
Ready to receive message
Waiting...
queue|<?xml version="1.0" encoding="UTF-8"?>
<bookstore id="2">
    <bookshelf>
        <category>Novels</category>
        <quantity>500</quantity>
    </bookshelf>
    <bookshelf>
        <category>Pictorials</category>
        <quantity>100</quantity>
    </bookshelf>
    <bookshelf>
        <category>Music</category>
        <quantity>400</quantity>
    </bookshelf>
</bookstore>
Closing connection
```

Line limit 100 Wrap

tMomMessageIdList

Propriétés du tMomMessageIdList

Famille du composant	Internet	
Fonction	Le composant **tMomMessageIdList** récupère la liste des ID des messages à partir de la file d'attente d'un système Middleware orienté Message et la passe au composant suivant.	
Objectif	Le **tMomMessageIdList** permet d'itérer les ID de certains messages. Il est généralement utilisé avec le **tMomInput**. Pour plus d'informations, consultez *tMomInput*.	
Basic settings	*Use existing connection*	Cochez cette case et, dans la liste qui s'affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies. Lorsqu'un Job contient un Job père et un Job fils, la liste qui s'affiche présente uniquement les composants de connexion du sous-job concerné.
	MQ Server	Sélectionnez dans cette liste le serveur MOM à utiliser. Les paramètres requis diffèrent légèrement en fonction du serveur sélectionné.
	Host/Port	Renseignez le nom de l'hôte (Host) et l'adresse IP du serveur MOM ainsi que le Port.
Websphere	*Channel*	La valeur par défaut est Channel.
	Queue Manager	Renseignez le gestionnaire de file(s) d'attente.
	Message Queue	Saisissez le nom de la file d'attente.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **CURRENT_MESSAGE_ID** : l'identifiant du message courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Il requiert un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Assurez-vous que le serveur Websphere correspondant est lancé. Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.	

Scénario associé

Pour un scénario associé, consultez *tMomInput*.

tMomOutput

Propriétés du tMomOutput

Famille de composant	Internet	
Fonction	Dépose un message sur la file d'attente d'un système Middleware orienté message afin qu'il soit récupéré de manière asynchrone.	
Objectif	Le **tMomOutput** permet de mettre en place des communications asynchrones via un serveur MOM.	
Basic settings	*Use existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component list** présente uniquement les composants de connexion du Job du même niveau.
	Failover	Cochez cette case pour afficher la table d'adresses du serveur. Dans cette table, vous pouvez définir plusieurs serveurs MQ pour le failover. (Disponible lorsque vous sélectionnez **ActiveMQ** dans la liste **MQ Server**.)
	MQ Server	Sélectionnez dans la liste le serveur MOM à utiliser. En fonction du serveur sélectionné, les paramètres diffèrent légèrement.
	Host/Port	Renseignez le nom de l'hôte ou l'adresse IP du serveur MOM dans le champ **Host**, ainsi que le **Port**.
	Username	Identifiant de connexion au serveur.
	Password	Mot de passe de connexion au serveur. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	To	Saisissez la destination du message, telle qu'attendue par le serveur ; le type est le nom du dossier cible doivent être indiqués, par exemple : queue/A ou topic/testtopic Notez que ce champ est sensible à la casse. (Disponible lorsque vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.)
	Message Type	Sélectionnez le type de message : **topic** ou **queue**. (Disponible lorsque vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.)
	Message Body Type	Sélectionnez le type de message : **Text**, **Bytes** ou **Map**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Lors de l'utilisation du composant **tMomOutput**, le schéma est en lecture seule mais changera en fonction du schéma d'entrée. Le serveur attend uniquement un schéma à une colonne pour contenir les **Messages**.

Websphere MQ	*Channel*	Renseignez ce champ avec le nom du canal avec lequel la connexion est établie. La valeur par défaut est *DC.SVRCONN*.
	Queue Manager	Programme système fournissant un conteneur logique pour la file de messages et responsable de transférer les données à d'autres gestionnaires de file via des canaux de messages. Renseignez ce champ avec le nom du gestionnaire de file(s) d'attente auquel la connexion est effectuée.
	Message Queue	File d'attente à partir de laquelle les applications de files d'attente de messages peuvent charger des messages et en récupérer. Renseignez ce champ avec le nom de la file de messages.
	Is using message id to set	Cochez cette case pour paramétrer les messages selon leur ID.
Advanced settings	*Additional options*	Définissez dans la table les options de transport pour le failover. Cette table est disponible uniquement si vous avez coché la case **Failover** dans la vue **Basic settings**. • **Option name** : nom de l'option. • **Option value** : valeur de l'option. Par exemple, saisissez *maxReconnectAttempts* dans le champ **Option name** puis *0* dans le champ **Option value** afin qu'aucune tentative ne soit effectuée pour récupérer la connexion lorsqu'aucun serveur n'est disponible pour le mode failover. Pour plus d'informations concernant les options de transport, consultez http://activemq.apache.org/failover-transport-reference.html (en anglais).
	Delivery Mode	Sélectionnez un mode de livraison supporté par JMS : **Not Persistent** : Ce mode de livraison ne nécessite pas que le message soit stocké de manière stable. **Persistent** : Ce mode de livraison nécessite que le message soit stocké de manière stable, en tant que partie de l'opération d'envoi effectuée par le client. Pour plus d'informations concernant les modes de livraison, consultez leChapitre 14 de MQSeries Using Java (en anglais). (Disponible lorsque vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.)
	Set Transacted	Cochez cette case pour activer l'option et gérer la session. Pour plus d'informations concernant ce paramètre, consultez leChapitre 14 de MQSeries Using Java (en anglais). Cocher cette case permet d'ignorer les paramètres de la liste **Acknowledgement Mode** dans la vue **Advanced settings** du **tMomInput**. Cette case est activée lorsque le serveur MQ est ActiveMQ ou JBoss Messaging.
	Set Jms Header	Cochez cette case pour envoyer les en-têtes JMS via le mapping des champs JMS vers les champs MQ Series sur le serveur MQ. Lorsque cette case est cochée, vous pouvez spécifier le nom de l'en-tête ainsi que sa valeur dans la (les) ligne(s) ajoutée(s) en cliquant sur le bouton **[+]** sous la table **Parameters**. Pour plus d'informations concernant l'utilisation des en-têtes JMS, consultez le Chapitre 12 de MQSeries Using Java (en anglais). (Disponible lorsque vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.)
	Set Jms Properties	Cochez cette case pour envoyer les propriétés JMS mappées dans les champs MQMD du serveur MQ. Lorsque cette case est cochée, vous pouvez spécifier le nom, le type et la valeur de la propriété dans la (les) ligne(s) ajoutée(s) en cliquant sur le bouton **[+]** sous la table **Parameters**. Pour plus d'informations concernant l'utilisation des propriétés JMS, consultez le Chapitre 12 de MQSeries Using Java.

		(Disponible lorsque vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.)
	Use format	Cochez cette case pour spécifier le format de message WebSphere dans le champ **WebSphere Message Format**. Le format par défaut est *MQSTR*. Pour plus d'informations concernant le format des messages WebSphere, consultez WebSphere Message Format (en anglais). (Disponible lorsque vous sélectionnez **Websphere MQ** dans la liste **MQ Server**.)
	Include Header	Cochez cette case pour définir l'en-tête du message : **MQRFH2 fixed Portion** : Cochez cette case et cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) pour spécifier la **Version**, l'encodage (**Encoding**), le **Format**, les **Flags** et les valeurs **NameValueCCSID** et **CodeCharSetId** pour la portion fixe de l'en-tête MQRFH2. **MCD folder** : Cochez cette case et cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) pour spécifier le nom et la valeur des propriétés décrivant le format du message. **JMS folder** : Cochez cette case et cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) pour spécifier le nom du champ et sa valeur pour le transport des champs d'en-têtes JMS et des propriétés JMSX. **USR folder** : Cochez cette case et cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) pour spécifier le nom du champ et sa valeur pour le transport des propriétés définies par l'application associée au message. Pour plus d'informations concernant les en-têtes MQRFH2, consultez http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r1m0/index.jsp?topic=%2Fcom.ibm.etools.mft.doc%2Faq06920_.htm. (Disponible lorsque vous sélectionnez **Websphere MQ** dans la liste **MQ Server**.)
	Set CipherSpec	Cochez cette case pour activer la liste **CipherSpec** dans laquelle vous pouvez spécifier le CipherSpec à utiliser avec le SSL WebSphere. Pour plus d'informations concernant CipherSpec, consultez Specifying CipherSpecs (en anglais). (Disponible lorsque vous sélectionnez **Websphere MQ** dans la liste **MQ Server**.)
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion au serveur MQ parmi celles prévues dans votre Job. La table **Dynamic settings** n'est disponible que si la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant nécessite un composant d'entrée ou un composant intermédiaire.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Assurez-vous que le serveur Websphere, JBoss ou ActiveMQ correspondant est bien lancé. Ce composant requiert l'installation des fichiers .jar liés. Pour plus d'informations sur l'installation de ces fichiers .jar manquants, consultez la section décrivant comment configurer le *Studio Talend*, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un exemple d'utilisation du **tMomOutput**, consultez *tMomInput*.

tMomRollback

Propriétés du tMomRollback

Famille de composant	Internet	
Fonction	Le **tMomRollback** annule la transaction sur le serveur MQ.	
Objectif	Ce composant évite le commit de transaction involontaire.	
Basic settings	*Component List*	Sélectionnez dans la liste le composant de connexion du Job en cours.
	MQ Server	Sélectionnez dans cette liste le serveur MOM à utiliser.
	Close Connection	Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en association avec des composants Mom, notamment avec le **tMomConnection** et le **tMomCommit**.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé au **tMomRollback**, consultez *Scénario : Annuler l'insertion de données dans des tables mère/fille* du composant **tMysqlRollback**.

tPOP

Propriétés du tPOP

Famille du composant	Internet	
Fonction	Le composant **tPOP** récupère un ou plusieurs e-mails à partir d'un protocole POP3 ou IMAP.	
Objectif	Le composant **tPOP** utilise le protocole POP ou IMAP pour se connecter au serveur de messagerie spécifié. Il récupère ensuite un ou plusieurs e-mails et écrit les informations récupérées dans un ou plusieurs fichiers définis. Les paramètres avancés (**Advanced settings**) permettent d'appliquer un ou plusieurs filtres à la sélection.	
Basic settings	*Host*	Renseignez le nom de l'hôte ou l'adresse IP du serveur de messagerie auquel vous voulez vous connecter.
	Port	Renseignez le numéro du port du serveur de messagerie.
	Username et *Password*	Renseignez les informations d'authentification à votre compte de messagerie : **Username** : entrez l'identifiant que vous utilisez pour accéder à votre boîte mail. **Password** : entrez le mot de passe de connexion à votre boîte mail.
	Output directory	Renseignez le chemin d'accès au dossier dans lequel vous voulez stocker les e-mails récupérés à partir de votre messagerie, ou cliquez sur **[...]** pour parcourir vos dossiers jusqu'à l'emplacement souhaité.
	Filename pattern	Appuyez sur **Ctrl+Espace** pour afficher la liste des modèles prédéfinis ou définissez manuellement la syntaxe du nom des fichiers cible. Les données des e-mails récupérés à partir du serveur de messagerie sont écrites dans ces fichiers.
	Retrieve all emails?	Par défaut, tous les messages présents sur le serveur de messagerie spécifié sont récupérés. Pour ne récupérer qu'un nombre défini de messages, décochez cette case et renseignez dans le champ **Number of emails to retrieve** : le nombre d'e-mails à récupérer. La collecte des e-mails se fait du plus récent au plus ancien.
	Delete emails from server	Cochez cette case si vous ne souhaitez pas conserver sur le serveur les e-mails récupérés. Pour les serveurs Gmail, cette option ne fonctionne pas avec le protocole POP3. Sélectionnez le protocole IMAP et vérifiez que le compte Gmail est configuré pour utiliser ce protocole.
	Choose the protocol	Sélectionnez à partir de la liste déroulante le protocole à utiliser pour récupérer les e-mails. Le protocole dépend de celui utilisé par votre messagerie. Si vous choisissez le protocole imap, vous aurez la possibilité de spécifier le dossier contenant les e-mails que vous souhaitez récupérer.
	Use SSL	Cochez cette case si votre serveur de messagerie utilise ce protocole de sécurisation des échanges. Cette option est obligatoire pour les utilisateurs de messagerie Gmail.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.

	Filter	Cliquez sur le bouton **[+]** pour ajouter des lignes de filtre afin de ne récupérer qu'une sélection d'e-mails :
		Filter item : sélectionnez l'un des types de filtre suivants à partir de la liste déroulante : **From** : les messages sont filtrés en fonction de l'adresse e-mail de provenance. **To** : les messages sont filtrés en fonction de l'adresse e-mail de destination. **Subject** : les messages sont filtrés en fonction du sujet. **Before date** : les messages sont filtrés en fonction de la date d'émission ou de réception ; ils doivent être antérieurs à la date indiquée. **After date** : les messages sont filtrés en fonction de la date d'émission ou de réception ; ils doivent être postérieurs à la date indiquée.
		Pattern : appuyez sur **Ctrl+Espace** pour afficher la liste des valeurs disponibles et sélectionnez la valeur à utiliser pour chacun des filtres.
	Filter condition relation	Sélectionnez le type de relation à appliquer entre les filtres : **and** : les conditions déterminées par les filtres s'ajoutent les unes aux autres, la recherche est plus restrictive. **or** : les conditions déterminées par les filtres sont indépendantes les unes des autres, la recherche est plus large.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_EMAIL** : le nombre d'e-mail reçus. Cette variable est une variable **After** et retourne un nombre entier. **CURRENT_FILE** : nom du fichier courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CURRENT_FILEPATH** : nom du fichier courant ainsi que son chemin d'accès. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant ne traite pas de flux de données, il peut donc être utilisé seul.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Lorsque la case **Use SSL** est cochée ou que le protocole **imap** est sélectionné, le **tPOP** ne fonctionne pas avec IBM Java 6.	

Scénario : Récupérer une sélection d'e-mails à partir d'un compte de messagerie

Ce scénario à un seul composant a pour but de récupérer un nombre prédéfini d'e-mails à partir d'un compte de messagerie.

- A partir de la **Palette**, cliquez-déposez un composant **tPOP** dans l'éditeur graphique,
- Double-cliquez sur le **tPOP** pour afficher la vue **Basic settings** et paramétrer le composant.
- Renseignez les informations de connexion au serveur de messagerie : le nom de l'hôte ou son adresse IP entre guillemets dans le champ **Host** et le numéro de port dans le champ **Port**.
- Saisissez ensuite les informations d'authentification de votre compte de messagerie. Dans le champ **Username**, saisissez entre guillemets votre identifiant et dans le champ **Password**, saisissez votre mot de passe. Dans cet exemple, le serveur de messagerie s'appelle *Free*.

- Dans le champ **Output directory**, sélectionnez manuellement le chemin d'accès ou cliquez sur **[...]** pour sélectionner le dossier de destination des e-mails récupérés sur le serveur.
- Dans le champ **Filename pattern**, définissez la syntaxe de nommage des fichiers de destination ou appuyez sur **Ctrl+Espace** pour afficher la liste des modèles prédéfinis. Dans cet exemple, la syntaxe est la suivante :

  `TalendDate.getDate("yyyyMMdd-hhmmss") + "_" + (counter_tPOP_1 + 1) + ".txt"` Les fichiers de destination, enregistrés au format texte, sont définis par leur date, heure et ordre de collecte.
- Décochez la case **Retrieve all emails?** pour ne récupérer qu'un nombre défini d'e-mails. Dans cet exemple, 10 messages sont récupérés.
- Cochez la case **Delete emails from server** pour supprimer les e-mails du serveur une fois qu'ils ont été récupérés en local.
- Sélectionnez un type de protocole à partir de la liste déroulante **Choose the protocol**. Le choix du protocole dépend de celui utilisé par votre fournisseur de messagerie. Certains fournisseurs comme *Gmail* utilisent les deux protocoles. Dans cet exemple, le protocole sélectionné est *pop3*.
- Enregistrez le Job puis appuyez sur **F6** pour l'exécuter.

Le composant **tPOP** a ainsi récupéré les 10 derniers messages du compte de messagerie spécifié.

Pour chaque e-mail récupéré, un fichier .txt a été créé dans le dossier *tPOP*. Chaque fichier contient les métadonnées d'envoi du message (expéditeur, destinataire, objet du message) ainsi que les données propres au message.

tREST

Propriétés du tREST

Famille du composant	Internet	
Fonction	Le composant **tREST** envoie des requêtes HTTP à un fournisseur de services Web REpresentational State Transfer (REST) et obtient les réponses correspondantes.	
Objectif	Le composant **tREST** sert de client d'un service Web REST. Il envoie des requêtes HTTP à un fournisseur de services Web et obtient les réponses correspondantes.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le composant **tREST** utilise toujours un schéma local (built-in) en lecture seule qui contient deux colonnes : - **Body** : stocke les résultats provenant du serveur - **ERROR_CODE** : stocke le code de statut HTTP lorsqu'une erreur survient pendant le processus d'invocation. Le sens spécifique des codes d'erreur est soumis aux définitions de votre fournisseur de services Web. Pour plus d'informations, consultez en.wikipedia.org/wiki/List_of_HTTP_status_codes (en anglais). Cliquez sur **Edit Schema** pour visualiser le schéma. *Modifier le schéma peut engendrer une perte de la structure du schéma et donc un échec du composant.*
	URL	Renseignez l'adresse URL du serveur Web REST à invoquer.
	HTTP Method	A partir de cette liste, sélectionnez une méthode HTTP qui décrit l'action souhaitée. Le sens spécifique des méthodes HTTP est soumis aux définitions de votre fournisseur de services Web. Ci-dessous, les définitions des méthodes HTTP généralement admises : - **GET** : récupère les données du côté serveur en fonction des paramètres donnés. - **POST** : crée et télécharge les données en fonction des paramètres donnés. - **PUT** : met à jour les données en fonction des paramètres donnés, ou crée les données si elles sont inexistantes. - **DELETE** : supprime les données en fonction des paramètres donnés.
	HTTP Headers	Saisissez les paires nom/valeur (**name**/**value**) pour les en-têtes HTTP afin de définir les paramètres de l'opération HTTP demandée. Pour connaître les définitions spécifiques d'en-têtes HTTP, consultez votre fournisseur de services Web REST. Pour plus d'informations, consultez http://en.wikipedia.org/wiki/List_of_HTTP_headers (en anglais).
	HTTP Body	Saisissez les données utiles (payload) à charger du côté serveur lorsque les actions POST ou PUT sont sélectionnées.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant..

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé comme client d'un service Web REST afin de communiquer avec un fournisseur de services Web REST. Il requiert un composant de sortie.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	L'exécution de JRE 1.6 est requise pour faire fonctionner ce composant.

Scénario : Créer et récupérer des données en invoquant un service Web REST

Ce scénario décrit un Job simple qui invoque un service Web REST afin de créer un nouvel enregistrement client du côté serveur et de récupérer les informations du client. Une fois exécuté, ce Job affiche les informations correspondantes dans la console de la vue **Run**.

- A partir de la **Palette**, cliquez-déposez les composants suivants dans l'espace de modélisation graphique : deux composants **tREST** et deux composants **tLogRow**. Renommez les deux composants **tREST** afin qu'ils décrivent au mieux les actions à effectuer.
- Connectez chaque composant **tREST** à un **tLogRow** à l'aide d'une connexion de type **Row** > **Main**.
- Connectez le premier composant **tREST** au second **tREST** à l'aide d'une connexion de type **Trigger** > **OnSubjobOk**.

- Double-cliquez sur le premier composant **tREST** afin d'afficher l'onglet **Basic settings** de sa vue **Component**.

- Renseignez le champ **URL** avec l'adresse URL du serveur Web que vous êtes sur le point d'invoquer. Notez que l'adresse URL fournie dans ce scénario est destinée essentiellement à la démonstration, ce n'est pas une adresse valide.

- A partir de la liste **HTTP Method**, sélectionnez **POST** afin d'envoyer une requête HTTP permettant de créer un nouvel enregistrement.

- Cliquez sur le bouton **[+]** afin d'ajouter une ligne au tableau **HTTP Headers**, et saisissez la paire nom-valeur (**name-value**) adéquate, soumise aux définitions de votre fournisseur de service, en indiquant le type de média des données utiles (payload) à envoyer du côté serveur. Dans ce scénario, saisissez *Content-Type* et *application/xml*. Pour plus d'informations concernant les types de média Internet, consultez http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.7 (en anglais).

- Saisissez dans le champ **HTTP Body** les données utiles (payload) à charger du côté serveur, à savoir *<Customer><name>Steven</name></Customer>* dans ce scénario, pour créer un enregistrement d'un nouvel utilisateur appelé *Steven*.

Si vous souhaitez inclure des guillemets double dans votre payload, assurez-vous d'utiliser un caractère d'échappement, la barre oblique inversée, avant chaque guillemet. Dans ce scénario par exemple, saisissez *<Customer><name>\"Steven\"</name></Customer>* si vous souhaitez mettre le prénom *Steven* entre guillemets double.

- Double-cliquez sur le second **tREST** afin d'afficher l'onglet **Basic settings** de sa vue **Component**.

- Renseignez le champ **URL** avec la même adresse URL que celle utilisée dans le premier composant **tREST**.

- A partir de la liste **HTTP Method**, sélectionnez **GET** afin d'envoyer une requête HTTP permettant de récupérer tous les enregistrements existants.

- Dans l'onglet **Basic settings** de la vue **Component** de chaque **tLogRow**, cochez la case **Print component unique name in front of each output row** (Afficher le nom unique du composant en face de chaque ligne de sortie) pour identifier au mieux les deux flux de sortie.

- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

  La console indique que le premier composant **tREST** envoie une requête HTTP côté serveur afin de créer un nouvel utilisateur appelé *Steven* et que le second composant **tREST** lit avec succès les données côté serveur comprenant les informations de l'utilisateur nouvellement créé.

tRSSInput

Propriétés du tRSSInput

Famille du composant	Internet	
Fonction	Le composant **tRSSInput** lit des flux RSS en utilisant des URL.	
Objectif	Le **tRSSInput** permet de suivre les billets postés sur certains sites en particulier afin d'en rassembler et organiser les informations rapidement et facilement pour votre utilisation.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Dans le contexte d'utilisation du **tRSSInput**, le schéma est composé de quatre colonnes : **TITLE**, **DESCRIPTION**, **PUBDATE** et **Link**. Les paramètres de titre sont en lecture seule, contrairement à leur type et leur longueur.
	RSS URL	Saisissez l'URL du flux RSS à lire.
	Read articles from	Lorsque cette case est cochée, le composant **tRSSInput** lit les articles du flux RSS à partir de la date renseignée en cliquant sur le bouton **[...]** situé à côté du champ **date time**.
	Max number of articles	Lorsque cette case est cochée, le composant **tRSSInput** lit autant d'articles que le nombre spécifié dans le champ **max amount**.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et les lignes contenant les erreurs seront ignorées.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que composant de début. Il requiert un composant de sortie.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Récupérer les billets fréquemment mis à jour sur un blog

Ce scénario Java à deux composants a pour but de récupérer les billets qui sont souvent mis à jour sur un blog, à partir d'un flux RSS local **Talend**, en utilisant le composant **tRSSInput**.

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l'éditeur graphique : le **tRSSInput** et le **tLogRow**.

2. Cliquez-droit sur le premier pour le connecter au second à l'aide d'un lien de type **Row Main**.

3. Dans l'éditeur graphique, sélectionnez le **tRSSInput** et cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tRSSInput**.

4. Saisissez l'URL d'accès au flux RSS. Dans ce scénario, le **tRSSInput** est associé au flux RSS **Talend** : *http://feeds.feedburner.com/Talend*.

5. Cochez/décochez les autres cases selon vos besoins. Dans ce scénario, on veut afficher les informations de deux articles datés du 20 juillet 2008.

6. Dans l'espace graphique, sélectionnez le **tLogRow** puis cliquez sur la vue **Component** pour en définir la configuration de base. Pour plus d'informations sur les propriétés du composant **tLogRow**, consultez *Propriétés du tLogRow*.

7. Enregistrez le Job puis appuyez sur **F6** pour l'exécuter.

```
Starting job RSSInput at 15:17 07/08/2008.
.-------------------------------------------------------------------------
|                                    #1. tLogRow_1
+-------------+-----------------------------------------------------------
| key         | value
+-------------+-----------------------------------------------------------
| TITLE       | Welcoming Jean-Luc Solans
| DESCRIPTION | Jean-Luc Solans joins Talend as VP of Strategy and Busines
| PUBDATE     | 24 Jul 2008 19:40:11 GMT
| LINK        | http://feeds.feedburner.com/~r/Talend/~3/344920575/
+-------------+-----------------------------------------------------------

.-------------------------------------------------------------------------
|                                #2. tLogRow_1
+-------------+-----------------------------------------------------------
| key         | value
+-------------+-----------------------------------------------------------
| TITLE       | Talend Open Profiler gets rave reviews
| DESCRIPTION | An interview and a product review of Talend Open Profiler.
| PUBDATE     | 23 Jul 2008 21:12:53 GMT
| LINK        | http://feeds.feedburner.com/~r/Talend/~3/343928056/
+-------------+-----------------------------------------------------------

Job RSSInput ended at 15:17 07/08/2008. [exit code=0]
```

Le composant **tRSSInput** a accédé au flux RSS du site de **Talend** à votre place et en a organisé les informations pour vous.

Deux billets sont affichés sur la console. Chacun d'eux a ses propres titre, description, date de publication et URL du flux RSS correspondante. Les blogs présentent le dernier billet en premier et vous pouvez faire défiler vers le bas pour lire les billets les plus anciens.

tRSSOutput

Propriétés du tRSSOutput

Famille du composant	Internet	
Fonction	Le composant **tRSSOutput** écrit des fichiers XML RSS_Feed ou Atom_Feed.	
Objectif	Le **tRSSOutput** permet de créer des fichiers XML contenant des flux RSS ou Atom.	
Basic settings	*File Name*	Chemin d'accès et nom du fichier à traiter, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	Append	Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.
	Mode	Sélectionnez RSS ou ATOM selon le flux que vous souhaitez générer.
	Channel (en mode RSS)	Les informations à renseigner dans cette zone concernent l'intégralité des données saisies, du site, etc. plutôt que des éléments particuliers. **Title** : Saisissez un titre significatif. **Description** : Entrez une description du contenu qui vous semble suffisamment détaillée. **Publication date** : Renseignez la date adéquate. **Link** : Saisissez l'URL appropriée.
	Feed (en mode ATOM)	**Title** : Saisissez un titre significatif. **Link** : Saisissez l'URL appropriée. **Id** : Saisissez l'URL valide correspondant au champ **Link**. **Update date** : Saisissez la date de mise à jour. **Author name** : Saisissez le nom de l'auteur.
	Optionnal Channel Elements	Renseignez dans le tableau les informations relatives aux métadonnées du flux RSS. Cliquez sur le bouton **[+]** pour ajouter de nouvelles lignes : **Element Name** : nom de la métadonnée. **Element Value** : contenu de la métadonnée.
	Schema type et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.

		• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Dans le contexte d'utilisation du **tRSSInput**, le schéma est composé de quatre colonnes : **TITLE**, **DESCRIPTION**, **PUBDATE** et **Link**. Les paramètres de titre sont en lecture seule, contrairement à leur type et leur longueur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit obligatoirement être relié à un composant de début ou un composant intermédiaire.	

Scénario 1 : Créer un flux RSS et stocker les fichiers sur un serveur FTP

Ce scénario Java a pour but de :

- créer un flux RSS pour des fichiers que vous souhaitez partager avec d'autres personnes, et
- stocker l'ensemble des fichiers sur un serveur FTP.

Ce scénario permet d'écrire un fichier XML de flux RSS à partir d'une table MySQL contenant des informations concernant des livres. Il ajoute également des liens vers les fichiers stockés sur un serveur FTP au cas où les utilisateurs voudraient avoir accès aux fichiers dans leur intégralité.

Déposer et relier les composants

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l'éditeur graphique : le **tMysqlInput**, le **tRSSOutput** et le **tFTPPut**.
2. Cliquez-droit sur le **tMysqlInput** et connectez-le au **tRSSOutput** à l'aide d'un lien de type **Row Main**.
3. Cliquez-droit sur le **tMysqlInput** et connectez-le au **tFTPPut** à l'aide d'un lien de type **OnSubjobOk**.

Définir la source des données d'entrée

1. Dans l'espace graphique, sélectionnez le **tMysqlInput**.

2. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tMysqlInput**.

3. Configurez le champ **Property type** en mode **Repository** puis cliquez sur le bouton **[...]** pour sélectionner la base de données adéquate dans la liste. Les détails de connexion ainsi que le schéma sont remplis automatiquement.

4. Dans le champ **Table Name**, saisissez le nom de votre table ou cliquez sur le bouton **[...]** et sélectionnez le nom de la table à partir de la liste. Dans ce scénario, la table Mysql de données saisies s'appelle "rss_talend" et le schéma comporte quatre colonnes, *TITLE*, *Description*, *PUBDATE*, et *LINK*.

5. Dans le champ **Query**, saisissez la requête de base de données en faisant particulièrement attention à ce que l'ordre des champs corresponde au schéma défini, ou cliquez sur **Guess Query**.

Créer un flux RSS

1. Dans l'espace graphique, sélectionnez le composant **tRSSOutput**.

2. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tRSSOutput**.

3. Dans le champ **File name**, utilisez le nom de fichier et son chemin d'accès définis par défaut ou parcourez vos dossiers pour définir votre propre fichier XML de sortie.

4. Sélectionnez le type d'encodage à partir de la liste **Encoding Type**.

5. Dans la zone **Mode**, sélectionnez **RSS**.

6. Dans le panneau **Channel**, saisissez un titre, une description, une date de publication et un lien définissant l'ensemble de vos données saisies.

7. Sélectionnez le type de schéma dans la liste **Schema** puis cliquez sur **Edit Schema** pour le modifier si nécessaire.

   Vous pouvez également cliquer sur **Sync Column** pour récupérer le schéma généré par le composant précédent.

8. Enregistrez votre Job puis appuyez sur **F6** pour en exécuter la première partie.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <rss version="2.0">
  - <channel>
      <title>library index</title>
      <description>grouping new books</description>
      <pubdate>2008-08-16 14:55:29</pubdate>
      <link>http://feeds.feedburner.com/Talend</link>
    - <item>
        <title>Book 1</title>
        <description>Summary of book1</description>
        <pubdate>2008-01-02</pubdate>
        <link>ftp://localhost/file_1.txt</link>
      </item>
    - <item>
        <title>Book 2</title>
        <description>Summary of book2</description>
        <pubdate>2008-01-02</pubdate>
        <link>ftp://localhost/file_2.txt</link>
      </item>
    - <item>
        <title>Book 3</title>
        <description>Summary of book3</description>
        <pubdate>2008-01-02</pubdate>
        <link>ftp://localhost/file_3.txt</link>
      </item>
    </channel>
  </rss>
```


Le composant **tRSSOutput** a créé en sortie un flux RSS au format XML pour les fichiers définis.

Ecrire des fichiers entiers sur un serveur FTP

Pour stocker des fichiers complets sur un serveur FTP :

1. Dans l'espace graphique, sélectionnez le **FTPPut** et cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tFTPPut**.

2. Saisissez le nom du serveur et le numéro de port dans les champs correspondants.
3. Renseignez vos détails de connexion dans les champs **Username** et **Password**.
4. Parcourez vos dossier jusqu'au répertoire local ou tapez manuellement le chemin d'accès dans le champ **Local directory**.
5. Entrez les détails du serveur de répertoire distant.
6. Sélectionnez le mode de transfert à partir de la liste **Transfer mode**.
7. Dans le panneau **Files**, cliquez sur le bouton **[+]** pour ajouter des lignes puis remplissez les masques de fichiers de tous les fichiers à copier sur le répertoire distant. Dans ce scénario, les fichiers à enregistrer sur le serveur FTP sont tous des fichiers texte.
8. Enregistrez votre Job puis appuyez sur **F6** pour l'exécuter.

Les fichiers définis sont copiés sur le serveur distant.

Scénario 2 : Créer un flux RSS contenant des métadonnées

Ce scénario Java à deux composants a pour but de créer un flux RSS contenant des métadonnées et de rediriger les informations obtenues dans un fichier XML de flux RSS en sortie.

Déposer et relier les composants

- A partir de la **Palette**, cliquez-déposez les composants **tRSSInput** et **tRSSOutput** dans l'éditeur graphique.

  Cliquez-droit sur le **tRSSInput** et connectez-le au **tRSSOutput** à l'aide d'un lien de type **Row Main**.

Configurer les composants

1. Double-cliquez sur le composant **tRSSInput** pour ouvrir sa vue **Basic settings** et définir sa configuration de base.

2. Saisissez l'URL du RSS_Feed. Dans ce scénario, le composant **tRSSOutput** mène au RSS_Feed *http://feeds.feedburner.com/Talend*.

3. Dans l'espace graphique, double-cliquez sur le composant pour ouvrir sa vue **Component** et définir sa configuration de base (**Basic settings**).

4. Dans le champ **File name**, utilisez le nom de fichier de sortie et son chemin d'accès définis par défaut ou parcourez vos dossiers pour définir votre propre fichier XML de sortie.

5. Sélectionnez le type d'encodage à partir de la liste **Encoding Type**.

6. Dans la zone **Mode**, sélectionnez **RSS**.

7. Dans le panneau **Channel**, saisissez un titre, une description, une date de publication et un lien définissant l'ensemble de vos données saisies.

8. Dans le tableau **Optional Channel Element**, définissez les métadonnées de votre flux RSS. Dans ce scénario, le flux comprend deux métadonnées : *copyright*, à laquelle est associée la valeur *tos*, et *language*, à laquelle est associée la valeur *en_us*.

9. Sélectionnez le type de schéma dans la liste **Schema** puis cliquez sur **Edit Schema** pour le modifier si nécessaire.

Vous pouvez également cliquer sur **Sync Column** pour récupérer le schéma généré par le composant précédent.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

Les fichiers définis sont copiés dans le fichier XML de sortie et les métadonnées apparaissent sous le nœud *<channel>*, en dessous des informations de base du flux RSS.

Scénario 3 : Créer un fichier XML de flux ATOM

Ce scénario Java décrit un Job à deux composants, qui génère des données et les écrit dans un fichier XML de flux ATOM.

Déposer et relier les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tFixedFlowInput** du groupe **Misc** et un **tRSSOutput** du groupe **Internet**.

2. Cliquez-droit sur le **tFixedFlowInput** et connectez-le au **tRSSOutput** à l'aide d'un lien **Row Main**.

3. Une fenêtre pop-up s'affiche et vous demande si vous souhaitez passer le schéma du **tRSSOutput** au **tFixedFlowInput**, cliquez sur **Yes** (Oui).

Configurer les composants

1. Dans l'espace de modélisation graphique, double-cliquez sur le **tFixedFlowInput** afin d'afficher sa vue **Component** et définir ses **Basic settings**.

2. Dans le champ **Number of rows**, laissez le paramètre par défaut, **1**, pour ne générer qu'une ligne de données.

3. Dans la zone **Mode**, laissez sélectionnée l'option **Use Single Table** et remplissez le tableau **Values**. Notez que le champ **Column** du tableau **Values** est rempli par les colonnes du schéma défini dans le composant.

4. Dans le champ **Value** du tableau **Values**, saisissez les données que vous souhaitez envoyer au composant suivant.

5. Dans l'espace de modélisation, double-cliquez sur le composant **tRSSOutput** afin d'afficher sa vue **Component** et définir ses **Basic settings**.

6. Cliquez sur le bouton **[...]** à côté du champ **File Name** afin de définir le répertoire et le nom du fichier XML de sortie.

7. Dans la zone **Mode**, sélectionnez **ATOM** pour générer un fichier XML de flux ATOM.

   *Comme le format du flux ATOM est strict, des informations par défaut sont requises afin de créer le fichier XML. Le schéma du **tRSSOutput** comprend donc des colonnes par défaut contenant ces informations. Ces colonnes par défaut sont grisées pour indiquer qu'elles ne doivent pas être modifiées. Si vous choisissez de modifier le schéma du composant, le fichier XML ATOM créé ne sera pas valide.*

8. Dans la zone **Feed**, saisissez un titre, un lien, un ID, la date de mise à jour, et le nom de l'auteur, afin de définir vos données comme un tout.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

tRSSOutput_ATOM.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <feed xmlns="http://www.w3.org/2005/Atom">
4    <title>Job using tRSSOutput</title>
5    <id>http://feeds.feedburner.com/Talend/</id>
6    <updated>2008-07-03T16:41:00Z</updated>
7    <link href="http://feeds.feedburner.com/Talend/"/>
8    <author>
9      <name>Talend Open Studio</name>
10   </author>
11   <entry>
12     <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
13     <title>Job using tRSSOutput in Talend Open Studio</title>
14     <link href="http://feeds.feedburner.com/Talend/"/>
15     <updated>2010-09-07T15:43:02Z</updated>
16     <summary>This is a Job made in Talend Open Studio and using
17   </entry>
18 </feed>
19
```

Le composant **tRSSOutput** crée un flux de sortie ATOM au format XML.

tSCPClose

Propriétés du tSCPClose

Famille de composant	Internet/SCP	
Fonction	Le composant **tSCPClose** ferme la connexion au protocole entièrement crypté.	
Objectif	Ce composant permet de fermer une connexion au protocole SCP.	
Basic settings	*Component list*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tSCPConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion SCP parmi celles prévues dans votre Job. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Ce composant est étroitement lié aux composants **tSCPConnection** et **tSCPRollback**. Il est généralement utilisé avec un composant **tSCPConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tSCPClose**, consultez *Gérer un fichier en utilisant SCP*.

tSCPConnection

Propriétés du tSCPConnection

Famille de composant	Internet/SCP	
Fonction	Le composant **tSCPConnection** ouvre une connexion SCP afin d'effectuer une transaction.	
Objectif	Le **tSCPConnection** permet d'ouvrir une connexion SCP afin d'effectuer des transferts sécurisés de fichiers en une seule transaction.	
Basic settings	*Host*	Adresse IP du serveur SCP.
	Port	Numéro du port d'écoute du serveur SCP.
	Username	Identifiant de connexion au serveur SCP.
	Authentication method	Méthode d'authentification au serveur SCP.
	Password	Mot de passe de connexion au serveur SCP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme sous-Job à un seul composant.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour un scénario associé, consultez *Gérer un fichier en utilisant SCP*.

tSCPDelete

Propriétés du tSCPDelete

Famille de composant	Internet/SCP	
Fonction	Le composant **tSCPDelete** supprime les fichiers sélectionnés d'un serveur distant par le biais d'un protocole entièrement crypté.	
Objectif	Le **tSCPDelete** permet de supprimer un fichier d'un serveur SCP donné.	
Basic settings	*Host*	Adresse IP du serveur SCP.
	Port	Numéro du port d'écoute du serveur SCP.
	Username	Identifiant de connexion au serveur SCP.
	Authentication method	Méthode d'authentification au serveur SCP.
	Password	Mot de passe de connexion au serveur SCP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Filelist	Nom ou chemin d'accès du fichier à supprimer.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **STATUS** : résultats d'exécution du composant. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d'autres composants.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSCPFileExists

Propriétés du tSCPFileExists

Famille de composant	Internet/SCP	
Fonction	Le composant **tSCPFileExists** vérifie si un fichier est présent sur le serveur distant, par le biais d'un protocole entièrement crypté.	
Objectif	Le **tSCPFileExists** permet de vérifier la présence d'un fichier sur le serveur SCP spécifié.	
Basic settings	*Host*	Adresse IP du serveur SCP.
	Port	Numéro du port d'écoute du serveur SCP.
	Username	Identifiant de connexion au serveur SCP.
	Authentication method	Méthode d'authentification au serveur SCP.
	Password	Mot de passe de connexion au serveur SCP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Remote directory	Chemin d'accès au répertoire du serveur distant.
	Filename	Nom du fichier à chercher.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **FILENAME** : nom du fichier traité. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **STATUS** : résultats d'exécution du composant. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d'autres composants.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Gérer un fichier en utilisant SCP

Dans ce scénario, un Job de six composants vérifie qu'un fichier défini existe dans un répertoire spécifié sur un serveur SCP distant, puis retourne un message indiquant le résultat de cette vérification. Si le fichier n'existe pas

dans ce répertoire, le Job met le fichier local sur le serveur SCP puis récupère le fichier depuis le serveur SCP vers un autre répertoire local défini afin de vérifier que le fichier a été correctement chargé.

Déposer et relier les composants

1. Ajoutez les six composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette** : **tSCPConnection**, **tSCPFileExists**, **tJava**, **tSCPPut**, **tSCPGet** et **tSCPClose**.

2. Reliez le **tSCPConnection** au **tSCPFileExists** à l'aide d'un lien **Trigger** > **On Subjob Ok**.

3. Reliez le **tSCPFileExists** au **tJava** à l'aide d'un lien **Trigger** > **On Component Ok** et au **tSCPPut** à l'aide d'un lien **Trigger** > **Run if**.

4. Reliez le **tSCPPut** au **tSCPGet** à l'aide d'un lien **Trigger** > **On Subjob Ok**.

5. Reliez le **tSCPGet** au **tSCPClose** à l'aide d'un lien **Trigger** > **On Subjob Ok**.

Configurer le composant tSCPConnection

1. Double-cliquez sur le **tSCPConnection** afin d'ouvrir sa vue **Basic settings**.

2. Saisissez le nom d'hôte ou l'adresse IP du serveur SCP ainsi que le numéro de port dans les champs correspondants.

3. Saisissez le nom d'utilisateur utilisé pour cette connexion dans le champ **Username**.

4. Dans la liste **Authentication method**, sélectionnez **Password**.

5. Cliquez sur le bouton **[...]** à côté du champ **Password** puis, dans la boîte de dialogue **[Enter a new password]** qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** pour sauvegarder les changements.

Configurer la vérification de l'existence du fichier

1. Double-cliquez sur le **tSCPFileExists** afin d'ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** puis, dans la liste **Component List**, sélectionnez la connexion que vous avez configurée.

3. Dans le champ **Filename**, saisissez le nom du fichier à vérifier entre guillemets doubles, *“testscp.txt”* dans cet exemple.

4. Double-cliquez sur le **tJava** afin d'ouvrir sa vue **Basic settings**.

5. Dans le champ **Code**, saisissez le code suivant :

```
System.out.println((String)globalMap.get("tSCPFileExists_1_FILENAME"));
System.out.println((String)globalMap.get("tSCPFileExists_1_STATUS"));
```

6. Sélectionnez l'onglet **Component** et cliquez sur le connexion **If** afin d'ouvrir sa vue **Basic settings**.

7. Dans le champ **Condition**, saisissez le code suivant :

```
!((String)globalMap.get("tSCPFileExists_1_STATUS")).equals("File exists.")
```

Ainsi, si le fichier n'existe pas dans le répertoire défini, le reste du Job s'exécute.

Configurer le composant tSCPPut

1. Double-cliquez sur le **tSCPPut** afin d'ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** puis, dans la liste **Component List**, sélectionnez la connexion que vous avez configurée.

3. Cliquez sur le bouton **[+]** afin d'ajouter une ligne à la table **Filelist** puis saisissez le chemin complet vers le fichier local que vous souhaitez charger dans la colonne **Source** correspondante, *"D:/SCPPut/testscp.txt"* dans cet exemple.

Configurer le composant tSCPGet

1. Double-cliquez sur le **tSCPGet** afin d'ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** puis, dans la liste **Component List**, sélectionnez la connexion que vous avez configurée.

3. Dans le champ **Local directory**, saisissez le chemin ou parcourez votre système vers le dossier dans lequel vous souhaitez stocker le fichier, *"D:/SCPGet"* dans cet exemple.

4. Dans la liste **Overwrite or Append**, sélectionnez **append**.

5. Cliquez sur le bouton **[+]** afin d'ajouter une ligne dans la table **Filelist** puis saisissez le chemin complet vers le fichier que vous souhaitez récupérer depuis le serveur SCP dans la colonne **Source** correspondante, *"./testscp.txt"* dans cet exemple.

Configurer le composant tSCPClose

1. Double-cliquez sur le **tSCPClose** afin d'ouvrir sa vue **Basic settings**.

2. Dans la liste **Component List**, sélectionnez la connexion que vous avez configurée.

Exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder le Job.

2. Assurez-vous que le fichier *testscp.txt* existe dans le répertoire local *D:/SCPPut* et qu'il n'existe ni dans le répertoire local *D:/SCPGet* ni dans le répertoire racine du serveur SCP.

3. Appuyez sur **F6** afin d'exécuter le Job.

Les informations concernant la vérification du fichier s'affichent dans la console **Run**.

4. Ouvrez le répertoire local *D:/SCPGet* et le répertoire racine du serveur SCP afin de constater la présence du fichier *testscp.txt*.

tSCPFileList

Propriétés du tSCPFileList

Famille de composant	Internet/SCP	
Fonction	Le composant **tSCPFileList** fait une opération d'itération sur les fichiers d'un répertoire du serveur distant, par le biais d'un protocole entièrement crypté.	
Objectif	Le composant **tSCPFileList** permet de lister les fichiers d'un serveur SCP spécifié.	
Basic settings	*Use an existing connection*	Cochez cette case et sélectionnez le composant de connexion adéquat dans la liste **Component list** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	Host	Spécifiez le nom d'hôte ou l'adresse IP du serveur SCP.
	Port	Spécifiez le numéro du port d'écoute du serveur SCP.
	Username	Saisissez l'identifiant de connexion au serveur SCP.
	Authentication method	Sélectionnez une des méthodes d'authentification au serveur SCP à partir de la liste déroulante : **Public key**, **Password**, ou **Keyboard Interactive**.
	Password	Saisissez le mot de passe de connexion au serveur SCP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Ce champ apparaît uniquement lorsque vous sélectionnez **Password** ou **Keyboard Interactive** dans la liste déroulante **Authentication Method**.
	Key Passphrase	Saisissez la phrase secrète pour la clé publique. Pour saisir la phrase secrète, cliquez sur le bouton **[...]** à côté du champ **Passphrase**, puis, dans la boîte de dialogue qui s'ouvre, saisissez la phrase secrète entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. Ce champ apparaît uniquement lorsque vous sélectionnez **Public key** dans la liste déroulante **Authentication method**.
	Private key	Parcourez votre système ou saisissez le chemin d'accès à la clé privée. Ce champ apparaît uniquement lorsque vous sélectionnez **Public key** dans la liste déroulante **Authentication method**.
	Command Separator	Spécifiez le caractère utilisé pour séparer les différentes commandes.
	Filelist	Cliquez sur le bouton **[+]** pour ajouter une commande ou une série de commandes pour lister les fichiers d'un répertoire du serveur SCP.
Global Variables	**CURRENT_LINE** : fichier en cours d'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de fichiers itérés jusqu'à présent. Cette variable est une variable **Flow** et retourne un entier.	

	STATUS : résultats d'exécution du composant. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé comme sous-Job à un seul composant ou en association avec d'autres composants. Ce composant est particulièrement pratique lorsqu'il est associé avec **tSCPGet** pour retrouver différents fichiers dans un répertoire, puisque **tSCPGet** ne supporte pas les caractères de remplacement.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé utilisant un protocole différent, consultez *Scénario : Itération sur un répertoire distant*.

tSCPGet

Propriétés du tSCPGet

Famille de composant	Internet/SCP	
Fonction	Le composant **tSCPGet** transfère les fichiers sélectionnés par le biais d'un protocole entièrement crypté via une connexion SCP.	
Objectif	Le **tSCPGet** permet de copier de fichiers du serveur SCP spécifié.	
Basic settings	*Host*	Adresse IP du serveur SCP.
	Port	Numéro du port d'écoute du serveur SCP.
	Username	Identifiant de connexion au serveur SCP.
	Authentication method	Méthode d'authentification au serveur SCP.
	Password	Mot de passe de connexion au serveur SCP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Local directory	Chemin d'accès du répertoire de destination.
	Overwrite or Append	Liste des options disponibles pour le transfert des fichiers.
	Filelist	Nom ou chemin d'accès du ou des fichier(s) à copier.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **STATUS** : résultats d'exécution du composant. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d'autres composants.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Gérer un fichier en utilisant SCP*.

tSCPPut

Propriétés du tSCPPut

Famille de composant	Internet/SCP	
Fonction	Le composant **tSCPPut** copie les fichiers sélectionnés sur un serveur SCP distant par le biais d'un protocole entièrement crypté.	
Objectif	Le **tSCPPut** permet de copier les fichiers dans le répertoire d'un serveur SCP distant.	
Basic settings	*Host*	Adresse IP d'un serveur SCP.
	Port	Numéro du port d'écoute du serveur SCP.
	Username	Identifiant de connexion au serveur SCP.
	Authentication method	Méthode d'authentification au serveur SCP.
	Password	Mot de passe de connexion au serveur SCP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Remote directory	Chemin d'accès au répertoire de destination.
	Filelist	Nom ou chemin d'accès au(x) fichier(s) à copier.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **STATUS** : résultats d'exécution du composant. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d'autres composants.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Gérer un fichier en utilisant SCP*.

tSCPRename

Propriétés du tSCPRename

Famille de composant	Internet/SCP	
Fonction	Le composant **tSCPRename** renomme des fichiers sur un serveur SCP distant.	
Objectif	Le **tSCPRename** permet de renommer un ou des fichier(s) sur le serveur SCP sélectionné.	
Basic settings	*Host*	Adresse IP du serveur SCP.
	Port	Numéro du port d'écoute du serveur SCP.
	Username	Identifiant de connexion au serveur SCP.
	Authentication method	Méthode d'authentification au serveur SCP.
	Password	Mot de passe de connexion au serveur SCP. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	File to rename	Nom ou chemin d'accès au fichier à renommer.
	Rename to	Saisissez le nouveau nom du fichier.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **STATUS** : résultats d'exécution du composant. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d'autres composants.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé utilisant un protocole différent, consultez *Scénario : Renommer un fichier situé sur un serveur FTP*.

tSCPTruncate

Propriétés du tSCPTruncate

Famille de composant	Internet/SCP	
Fonction	Le composant **tSCPTruncate** supprime toutes les données d'un fichier via une connexion SCP.	
Objectif	Le **tSCPTruncate** permet de supprimer les données d'un ou plusieurs fichier(s) d'un serveur SCP.	
Basic settings	*Host*	Adresse IP du serveur SCP.
	Port	Numéro du port d'écoute du serveur SCP.
	Username	Identifiant de connexion au serveur SCP.
	Authentication method	Méthode d'authentification au serveur SCP.
	Password	Mot de passe de connexion au serveur SCP.
	Remote directory	Chemin d'accès au fichier de destination.
	Filelist	Nom ou chemin d'accès au(x) fichier(s) à vider.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier. **STATUS** : résultats d'exécution du composant. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que sous-Job à un seul composant mais il peut aussi être utilisé en association avec d'autres composants.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSendMail

Propriétés du tSendMail

Famille de composant	Internet	
Fonction	Le composant **tSendMail** envoie des messages et toute autre pièce jointe aux destinataires spécifiés.	
Objectif	Le **tSendMail** permet de prévenir des destinataires d'un état particulier d'un Job ou de possibles erreurs.	
Basic settings	*To*	Adresse e-mail du destinataire principal.
	From	Adresse e-mail du serveur d'envoi.
	Show sender's name?	Cochez cette case pour définir le nom d'expéditeur à afficher dans les messages. **Sendername** : renseignez le nom de l'expéditeur.
	Cc	Adresse e-mail des destinataires en copie.
	Bcc	Adresse e-mail des destinataires en copie cachée.
	Subject	Objet du mail.
	Message	Contenu du message. Appuyez sur **Ctrl+Espace** pour afficher la liste des variables disponibles.
	Die if the attachment file doesn't exist	Décochez cette case pour que les messages soient envoyés même si aucune pièce jointe n'est attachée.
	Attachment / File and Content Transfer Encoding	Cliquez sur le bouton **[+]** pour ajouter autant de lignes que nécessaire dans lesquelles vous pouvez ajouter des masques de fichier ou des chemins d'accès aux fichiers à envoyer dans l'e-mail, s'il y en a. Deux options sont disponibles pour l'encodage du contenu à transférer : **Default** et **Base64**.
	Other Headers	Cliquez sur le bouton **[+]** pour ajouter des lignes de filtre : **Key** : saisissez la clé de tout en-tête différent de l'en-tête standard. **Value** : saisissez la valeur correspondante à la clé.
	SMTP Host et Port	Adresse IP du serveur SMTP utilisé pour l'envoi d'e-mails.
	SSL Support	Cochez cette case pour authentifier le serveur auprès du client via un protocole sécurisé SSL.
	STARTTLS Support	Cochez cette case pour authentifier le serveur auprès du client via un protocole sécurisé STARTTLS.
	Importance	Sélectionnez dans la liste déroulante le niveau de priorité de votre message.
	Need authentication / Username and Password	Cochez la case d'authentification et renseignez le nom de l'utilisateur (username) et son mot de passe (password), si cela est nécessaire pour accéder au service. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et les lignes contenant les erreurs seront ignorées.
Advanced settings	*MIME subtype from the 'text' MIME type*	Sélectionnez dans la liste déroulante la forme à attribuer au texte du message.

	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que sous-Job mais peut aussi être utilisé comme composant de sortie ou de fin de Job. Il peut être connecté aux autres composants via des liens Row ou Iterate.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Envoyer un e-mail lors d'une erreur

Ce scénario est un Job composé de trois composants permettant d'envoyer un e-mail à des destinataires spécifiques lorsqu'une erreur se produit.

- A partir de la **Palette**, cliquez-déposez les composants suivants dans l'espace de modélisation : **tFileInputDelimited**, **tFileOutputXML** et **tSendMail**.
- Cliquez-droit sur le composant **tFileInputDelimited** et reliez-le au composant **tFileOutputXML** à l'aide d'un lien de type **Row** > **Main**.
- Cliquez-droit sur le composant **tFileInputDelimited** et reliez-le au composant **tSendMail** à l'aide d'un lien de type **Trigger** > **OnSubjobError**.
- Double-cliquez sur le composant **tFileInputDelimited** pour afficher l'onglet **Basic settings** du composant et configurer ses paramètres de base.

- A partir de la liste **Property type**, sélectionnez :

  -**Repository** si vous avez déjà stocké les métadonnées de votre fichier d'entrée dans le **Repository**, les champs suivants sont alors renseignés automatiquement avec les informations stockées, ou

  -**Built-in** si aucune métadonnée n'est stockée pour votre fichier d'entrée ; vous devez alors renseigner manuellement les champs suivants.

  Dans cet exemple, c'est le mode **Built-in** qui est utilisé.

- Dans le champ **File Name**, renseignez le chemin d'accès ou cliquez sur le bouton **[...]** pour parcourir vos dossiers jusqu'au fichier à traiter. Ce champ est obligatoire.

  Dans cet exemple, le fichier à traiter est *customer_mail.csv*.

- Définissez le séparateur de lignes dans le champ **Row separator** afin d'identifier la fin d'une ligne. Puis définissez le séparateur de champs dans **Field separator** pour délimiter les champs composant une ligne.

  Les séparateurs utilisés dans ce scénario sont des séparateurs de champs et de lignes classiques de fichier CSV.

- Dans ce scénario, l'en-tête (**Header**), le pied de page (**Footer**) et la limite de lignes lues (**Limit**) n'ont pas besoin d'être définis.

- Sélectionnez l'option **Built-in** à partir de la liste déroulante **Schema** et renseignez manuellement les informations concernant le fichier délimité dans les champs correspondants.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour afficher la boîte de dialogue permettant de définir la structure des données. Dans ce scénario, le schéma comporte quatre colonnes : *Id*, *Name*, *Address* et *IdState*.

- Cliquez sur le bouton **[+]** pour ajouter autant de lignes que nécessaire à la structure des données. Dans cet exemple, il n'y a qu'une seule colonne dans le schéma : *xmlStr*.

- Cliquez sur **OK** pour valider vos changements et fermer la boîte de dialogue.

  Si vous avez préalablement stocké votre schéma dans le répertoire **Metadata** sous **File delimited**, sélectionnez l'option **Repository** dans la liste déroulante **Schema** puis cliquez sur le bouton **[...]** à côté du champ pour afficher la boîte de dialogue **[Repository Content]** à partir de laquelle vous pouvez sélectionner un schéma pertinent. Cliquez sur **OK** pour fermer la boîte de dialogue et remplir les champs automatiquement avec les métadonnées contenues dans le schéma.

  Pour plus d'informations sur le stockage des schémas de métadonnées dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

- Assurez-vous que la case **Skip empty rows** est cochée afin d'ignorer les lignes vierges.

- Double-cliquez sur le composant **tFileOutputXML** pour afficher l'onglet **Basic settings** du composant et configurer ses paramètres de base.

- Cliquez sur le bouton **[...]** à côté du champ **File Name** pour sélectionner le répertoire de destination du fichier et saisissez le nom du fichier. Dans ce scénario, appelez le fichier *customer_data.xml*.
- Dans le champ **Row tag**, saisissez entre guillemets le nom que vous souhaitez donner à la balise contenant les données.
- Cliquez sur **Edit schema** et assurez-vous que le schéma est récupéré à partir du schéma d'entrée. Si nécessaire, cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.
- Double-cliquez sur le composant **tSendMail** pour afficher l'onglet **Basic settings** du composant et configurer ses paramètres de base.

- Saisissez les adresses e-mail des destinataires dans le champ **To** et les champs **Cc** et **Bcc** selon vos préférences, ainsi que celle de l'expéditeur dans le champ **From**.
- Cochez la case **Show sender's name?** puis renseignez le nom de l'expéditeur dans le champ **Sendername** pour que votre message ait une meilleure visibilité auprès de vos destinataires.
- Renseignez l'objet du message dans le champ **Subject**.
- Saisissez dans le champ **Message** le message contenant le code d'erreur produit à l'aide de la variable globale correspondante. Accédez à la liste des variables en appuyant sur **Ctrl+Espace**.
- Ajoutez des pièces jointes et des informations sur les en-têtes, si nécessaire, dans les champs respectifs **Attachments** et **Other headers**.
- Saisissez les informations SMTP dans les champs **SMTP host** et **SMTP port**.
- Enregistrez votre Job puis appuyez sur **F6** pour l'exécuter.

Dans ce scénario, le fichier contenant les données à transférer vers une sortie XML ne peut être trouvé. Le composant **tSendMail** s'exécute lorsque cette erreur est rencontrée et envoie un mail de notification au destinataire spécifié.

tSetKerberosConfiguration

Propriétés du tSetKerberosConfiguration

Famille de composant	Internet	
Fonction	Le composant **tSetKerberosConfiguration** est conçu pour configurer l'authentification via Kerberos afin d'améliorer la sécurité des communications réseau. Pour plus d'informations concernant le protocole Kerberos, consultez http://www.kerberos.org (en anglais).	
Objectif	Le **tSetKerberosConfiguration** vous permet de saisir les informations d'authentification via Kerberos.	
Basic settings	*KDC Server*	Adresse du centre de distribution de clés (Key Distribution Center, KDC).
	Realm	Nom du royaume (realm) de Kerberos.
	Username and Password	Informations d'authentification de Kerberos. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour rassembler les métadonnées de traitement du Job au niveau du Job et au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que sous-job et est utilisé avec le **tSoap**.	
Limitation	Le composant **tSetKerberosConfiguration** ne fonctionne pas avec une JVM IBM.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSetKeystore

Propriétés du tSetKeystore

Famille de composant	Internet	
Fonction	Le composant **tSetKeystore** soumet des données d'authentification d'un TrustStore avec ou sans Keystore pour valider la connexion SSL.	
Objectif	Ce composant vous permet de définir le type des données d'authentification entre **PKCS 12** et **JKS**.	
Basic settings	*TrustStore type*	Sélectionnez le type TrustStore à utiliser. Il peut être **PKCS 12** ou **JKS**.
	TrustStore file	Saisissez le chemin d'accès et le nom du fichier TrustStore, ou parcourez votre répertoire jusqu'au TrustStore contenant la liste des certificats approuvés par le client.
	TrustStore password	Saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore.
	Need Client authentication	Cochez cette case pour valider les données keystore. Vous devez ensuite compléter trois champs : - **KeyStore type** : sélectionnez le type de keystore à utiliser. Il peut être **PKCS 12** ou **JKS**. - **KeyStore file** : saisissez le chemin d'accès et le nom du fichier Keystore, ou parcourez votre répertoire jusqu'au fichier contenant les données keystore. - **KeyStore password** : saisissez le mot de passe keystore.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé en standalone.	
Connections	Liens de sortie (de ce composant à un autre) : **Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Trigger** : Run if, On Subjob Ok, On Component Ok, On Component Error. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	

Scénario : Extraire des informations client d'un fichier WSDL privé

Ce scénario décrit un Job comprenant trois composants, qui se connecte à un fichier WSDL privé, pour extraire des informations client.

Le fichier WSDL utilisé dans ce Job accède au service Web correspondant via le protocole SSL. Le code correspondant dans le fichier est le suivant :

```
<wsdl:port name="CustomerServiceHttpSoap11Endpoint"
binding="ns:CustomerServiceSoap11Binding">
            <soap:address location="https://192.168.0.22:8443/axis2/services/
CustomerService.CustomerServiceHttpSoap11Endpoint/"/>
        </wsdl:port>
```

Saisissez le code suivant dans le fichier *server.xml* de Tomcat :

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               clientAuth="true" sslProtocol="TLS"
     keystoreFile="D:/server.keystore" keystorePass="password"
     keystoreType="JKS"
     truststoreFile="D:/server.p12" truststorePass="password"
     truststoreType="PKCS12"
    />
```

Vous avez besoin de fichiers Keystore pour vous connecter à ce fichier WSDL. Pour reproduire ce Job, procédez comme suit :

- Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tSetKeystore**, **tWebService**, et **tLogRow**.

- Cliquez-droit sur le composant **tSetKeystore** pour ouvrir son menu contextuel.
- Dans ce menu, sélectionnez **Trigger** > **On Subjob Ok** afin de relier ce composant au **tWebService**.
- Cliquez-droit sur le **tWebService** pour ouvrir son menu contextuel.
- Dans ce menu, sélectionnez **Row** > **Main** pour connecter ce composant au **tLogRow**.
- Double-cliquez sur le **tSetKeystore** pour ouvrir sa vue **Basic settings** et définir ses propriétés.

- Dans la liste **TrustStore type**, sélectionnez **PKCS12**.
- Dans le champ **TrustStore file**, parcourez votre répertoire jusqu'au fichier TrustStore. Ici, le fichier est *server.p12*.
- Dans le champ **TrustStore password**, saisissez le mot de passe pour ce fichier TrustStore. Dans cet exemple, le mot de passe est *password*.
- Cochez la case **Need Client authentication** afin d'activer les champs de configuration Keystore.
- Dans la liste **KeyStore type**, sélectionnez **JKS**.
- Dans le champ **KeyStore file**, parcourez votre répertoire jusqu'au fichier Keystore correspondant. Ici, le fichier est *server.keystore.*
- Double-cliquez sur le composant **tWebService** pour ouvrir l'éditeur du composant, ou sélectionnez le composant dans l'espace de modélisation graphique et dans sa vue **Basic settings**, cliquez sur le bouton **[...]** à côté du champ **Service configuration**.

- Dans le champ **WSDL**, parcourez votre répertoire jusqu'au fichier WSDL privé à utiliser. Dans cet exemple, ce fichier est *CustomerService.wsdl*.

- Cliquez sur le bouton **Refresh** à côté du champ **WSDL** afin de récupérer la description WSDL et l'afficher dans les champs qui suivent.
- Dans la liste **Port Name**, sélectionnez le port que vous souhaitez utiliser, *CustomerServiceHttpSoap11Endpoint* dans cet exemple.
- Dans la liste **Operation**, sélectionnez le service que vous souhaitez utiliser. Dans cet exemple, le service sélectionné est *getCustomer(parameters):Customer*.
- Cliquez sur **Next** pour ouvrir une nouvelle vue dans l'éditeur.

Dans le panneau de droite de la vue **Input mapping**, le paramètre d'entrée du service s'affiche automatiquement. Cependant, vous pouvez ajouter d'autres paramètres si vous sélectionnez **[+] parameters** et que vous cliquez sur le bouton **[+]** en haut, afin d'afficher la boîte de dialogue **[Parameter Tree]** dans laquelle vous pouvez sélectionner n'importe lequel des paramètres listés.

Le service Web de cet exemple n'a qu'un seul paramètre d'entrée, *ID*.

- Dans la colonne **Expression** de la ligne *parameters.ID*, saisissez entre guillemets l'ID du client qui vous intéresse. Dans cet exemple, l'ID est *A00001*.
- Cliquez sur **Next** pour ouvrir une nouvelle vue dans l'éditeur.

Dans la liste **Element** à gauche de la vue s'affiche automatiquement le paramètre de sortie du service Web. Cependant, vous pouvez ajouter d'autres paramètres si vous sélectionnez **[+] parameters** et que vous cliquez sur le bouton **[+]** en haut afin d'afficher la boîte de dialogue **[Parameter Tree]** dans laquelle vous pouvez sélectionner n'importe lequel des paramètres affichés.

Le service Web dans cet exemple contient quatre paramètres de sortie : *return.address*, *return.email*, *return.name* et *return.phone*.

Vous devez créer une connexion entre le paramètre de sortie du service Web défini et le schéma du composant de sortie. Pour ce faire :

- Dans le panneau à droite de la vue, cliquez sur le bouton **[...]** à côté du champ **Edit Schema** afin d'ouvrir une boîte de dialogue dans laquelle vous pouvez définir le schéma de sortie.

- Dans l'éditeur de schéma, cliquez sur le bouton **[+]** pour ajouter quatre colonnes au schéma de sortie.
- Cliquez dans chaque colonne et saisissez les nouveaux noms : *Name*, *Phone*, *Email* et *Address* dans cet exemple. Cela vous permettra de récupérer les informations client qui vous intéressent.
- Cliquez sur **OK** pour valider les modifications et fermer la boîte de dialogue du schéma.
- Dans la liste **Element** à droite de l'éditeur, glissez chaque paramètre dans le champ correspondant à la colonne que vous avez définie dans l'éditeur de schéma.

S'il est disponible, vous pouvez utiliser le bouton **Auto map!**, situé en bas à gauche de l'interface, afin d'effectuer l'opération de mapping automatiquement.

- Cliquez sur **OK** pour valider vos modifications et femer l'éditeur.
- Dans l'espace de modélisation graphique, double-cliquez sur le composant **tLogRow** afin d'ouvrir sa vue **Basic settings** et définir ses propriétés.
- Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

Les informations du client dont l'ID est *A00001* sont retournées et affichées dans la console du *Studio Talend*.

```
.-------------+--------------+-------------+-------------------------------------------.
|                                   tLogRow_2                                           |
|=------------+--------------+-------------+-------------------+----------------------=|
|name         |phone         |email        |address                                    |
|=------------+--------------+-------------+-------------------------------------------=|
|Rose Gonzalez|(512) 757-9000|rose@edge.com|313 Constitution Place Austin, TX 78767 USA|
'-------------+--------------+-------------+-------------------------------------------'
[statistics] disconnected
Job tSetKeyStore ended at 19:01 24/11/2010. [exit code=0]
```

tSetProxy

Propriétés du tSetProxy

Famille de composant	Internet	
Fonction	Le composant **tSetProxy** est conçu pour configurer un proxy.	
Objectif	Le **tSetProxy** vous permet de saisir les informations de configuration de votre proxy.	
Basic settings	*Proxy type*	Liste des protocoles de proxy.
	Proxy host	Adresse du serveur du proxy.
	Proxy port	Numéro du port du proxy.
	Proxy user	Identifiant d'authentification pour le proxy.
	Proxy password	Mot de passe d'authentification pour le proxy. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	None proxy hosts	Les hôtes auxquels vous devez vous connecter directement et non via le serveur de proxy. La valeur peut être une liste d'hôtes, séparés par le caractère |. Le caractère de remplacement (*) peut être utilisé pour la mise en correspondance, par exemple : "192.168.0.* | localhost".
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le **tSetProxy** peut être utilisé en sous-job indépendant, mais peut également être déployé avec d'autres composants de la famille Internet.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tSocketInput

Propriétés du tSocketInput

Famille de composant	Internet	
Fonction	Le composant **tSocketInput** ouvre un port de socket et permet d'écouter les données envoyées.	
Objectif	Le **tSocketInput** est un composant d'écoute, permettant de transmettre des données via un port défini.	
Basic settings	*Host name*	Nom ou adresse IP du serveur Hôte.
	Port	Port d'écoute à ouvrir.
	Timeout	Nombre de secondes avant fermeture du socket du port d'écoute.
	Uncompress	Cochez cette case pour extraire les données si nécessaire.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Row separator	Chaîne (ex : "\n"sous Unix) utilisée pour séparer les lignes.
	Escape Char	Caractère de la ligne à éviter.
	Text enclosure	Caractères utilisés pour entourer le texte.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant ouvre un point d'accès au poste de travail ou au serveur. Ce composant est un composant de début de Job et s'arrête uniquement après expiration.
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Transférer des données vers un port d'écoute

Le scénario suivant décrit un double Job dont le but est de transférer des données via un port d'écoute. Le composant Socket peut aussi permettre de contrôler la communication entre les serveurs ne pouvant pas communiquer directement entre eux.

Créez deux Jobs : le premier (*SocketInput*) ouvre un port d'écoute et attend que les données soient envoyées. Le deuxième (*SocketOutput*) transmet des données délimités d'un fichier vers le numéro de port correspondant au port d'écoute.

Déposer et relier les composants

1. Dans le premier Job, cliquez et déposez les composants suivants dans l'espace numérique de travail : **tSocketInput** et **tLogRow**. Reliez à l'aide d'une connexion **Row**> **Main**

2. Dans le deuxième Job, cliquez et déposez les composants suivants de la **Palette** : **tFileInputDelimited** et **tSocketOutput**. Reliez à l'aide d'une connexion **Row**> **Main**.

Configurer les Jobs

1. Dans le second Job, sélectionnez **tFileInputDelimited** et dans l'onglet **Basic settings** de la vue **Component**, configurez les paramètres d'accès au fichier d'entrée.

2. Dans le champ **File Name**, renseignez le chemin d'accès au fichier. Renseignez les séparateurs de lignes et de champs respectivement dans les champs **Row** et **Field separators**, ainsi que l'en-tête dans le champ **Header**.

3. Décrivez le schéma des données à passer au composant **tSocketOutput**.

4. Sélectionnez le composant **tSocketOutput** et configurez les paramètres de l'onglet **Basic settings** de la vue **Component**.

5. Dans le champ **Host**, renseignez l'adresse IP et dans le champ **Port**, renseignez le numéro du port vers lequel les données seront transférées.

6. Dans le champ **Retry**, définissez le nombre d'essais à effectuer et dans le champ **Timeout** renseignez la durée (en secondes) avant que le Job ne s'arrête automatiquement.

7. Maintenant, configurez les paramètres du premier Job (*SocketInput*) contenant le composant **tSocketInput**.

8. Dans le champ **Host**, renseignez l'adresse IP et dans le champ **Port**, renseignez le numéro du port d'écoute vers lequel les données sont transférées.

9. Dans le champ **Timeout**, définissez la durée (en secondes) avant que le Job ne s'arrête automatiquement.

10. Modifiez le schéma afin qu'il corresponde en tout point ou en partie au schéma du deuxième Job.

Exécuter les Jobs

1. Appuyez sur **F6** pour exécuter le Job *SocketInput* en premier afin d'ouvrir le port d'écoute et de le préparer à recevoir les données.

2. Le résultat s'affiche dans la vue **Run**, ainsi que les informations d'ouverture du socket.

```
Starting job SocketInput at 17:53 04/02/2008.
socket connected
AL|Alabama|Montgomery|Birmingham
AK|Alaska|Juneau|Anchorage
AZ|Arizona|Phoenix|Phoenix
AR|Arkansas|Little Rock|Little Rock
CA|California|Sacramento|Los Angeles
CO|Colorado|Denver|Denver
CT|Connecticut|Hartford|Bridgeport
DE|Delaware|Dover|Wilmington
FL|Florida|Tallahassee|Jacksonville
GA|Georgia|Atlanta|Atlanta
HI|Hawaii|Honolulu|Honolulu
ID|Idaho|Boise|Boise
IL|Illinois|Springfield|Chicago
IN|Indiana|Indianapolis|Indianapolis
```

Avant que le Job ne s'arrête, lancez l'autre Job (SocketOutput) afin de transmettre les données.

tSocketOutput

Propriétés du tSocketOutput

Famille de composant	Internet	
Fonction	Le composant **tSocketOutput** écrit des données sur un port d'écoute.	
Objectif	Le **tSocketOutput** transfert des données d'un flux d'entrée vers le port d'écoute du socket.	
Basic settings	*Host*	Nom ou adresse IP du serveur Hôte.
	Port	Port d'écoute à ouvrir
	Compress	Cochez cette case pour archiver les données si nécessaire.
	Retry times	Nombre d'essais avant que le Job ne soit mis en échec.
	Timeout	Nombre de secondes avant fermeture du port d'écoute.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	Row separator	Chaîne (ex : "\n" sous Unix) séparant les lignes.
	Escape Char	Caractère de la ligne à éviter.
	Text enclosure	Caractères utilisés pour entourer le texte.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant ouvre un point d'accès au poste de travail ou au serveur. Ce composant est un composant de début de Job et s'arrête uniquement après expiration.
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario associé

Pour un exemple d'utilisation du **tSocketOutput**, consultez *Scénario : Transférer des données vers un port d'écoute*.

tSOAP

Propriétés du tSOAP

Famille de composant	Internet	
Fonction	Le composant **tSOAP** envoie le message SOAP défini avec les paramètres donnés au service Web invoqué et retourne la valeur comme définie, à partir des paramètres donnés.	
Objectif	Ce composant appelle une méthode via un service Web afin de récupérer les valeurs des paramètres définies dans l'éditeur du composant.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Ce composant utilise toujours un schéma local (built-in) en lecture seule qui contient par défaut trois colonnes de type **String** : - **Header** : stocke l'en-tête du message SOAP de la réponse donnée côté serveur. - **Body** : stocke le corps du message SOAP de la réponse donnée côté serveur. - **Fault** : stocke les informations d'erreur lorsqu'une erreur survient pendant le traitement du message SOAP. Cliquez sur **Edit Schema** pour visualiser le schéma. *Modifier le schéma peut engendrer une perte de la structure du schéma et donc un échec du composant.*
	Use NTLM	Cochez cette case si vous utilisez un protocole d'authentification NTLM. **Domain** : Nom de domaine du client.
	Need authentication	Cochez la case d'authentification et renseignez le nom de l'utilisateur dans le champ **Username** et son mot de passe dans le champ **Password**, si cela est nécessaire pour accéder au service. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use http proxy	Cochez cette case si vous vous connectez derrière un proxy et renseignez les informations correspondantes.
	Trust server with SSL	Cochez cette case pour authentifier le serveur auprès du client via un protocole SSL et renseignez les champs correspondants. **TrustStore file** : saisissez le chemin d'accès et le nom du fichier TrustStore contenant la liste des certificats approuvés par le client. **TrustStore password** : saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore.
	ENDPOINT	Saisissez l'URL du serveur Web invoqué.
	SOAP action	Saisissez l'URL de l'en-tête HTTP SOAPAction à utiliser pour identifier le but de la requête HTTP SOAP.
	SOAP version	Sélectionnez la version du système SOAP que vous utilisez.

		L'enveloppe SOAP requise varie selon les versions.
	Use a message from the input schema	Cochez cette case pour lire un message SOAP du composant précédent et l'envoyer au service Web invoqué. Lorsque cette case est cochée, le champ **SOAP message** devient une liste déroulante, vous permettant de sélectionner le type de colonne **Document** afin de lire un fichier d'entrée XML. *Il est logique d'utiliser cette option lorsque le composant **tSOAP** est relié à un composant d'entrée dont le schéma contient une colonne de type **Document** afin de lire un message SOAP valide.*
	Output in Document	Cochez cette case afin d'écrire le message de réponse au format XML.
	SOAP message	Saisissez le message SOAP à envoyer au service Web invoqué. Les variables globales et de contexte peuvent être utilisées lorsque vous écrivez un message SOAP. Pour plus d'informations concernant les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.
Advanced settings	*Use Kerberos*	Cochez cette case pour sélectionner un composant **tSetKerberosConfiguration** dans la liste **Kerberos configuration**. Le lien **Trigger** > **OnSubjobOk** du composant **tSetKerberosConfiguration** doit être utilisé lors d'une connexion au **tSOAP**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en tant que composant d'entrée ou intermédiaire.	
Connections	Liens de sortie (de composant à un autre) : **Row** : Main, Iterate. **Trigger** : Run if, On Component Ok, On Component Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Iterate **Trigger** : Run if, On Component Ok, On Component Error. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Récupérer le nom d'un pays en utilisant un Service Web

Ce scénario décrit un Job à deux composants utilisant un Service Web pour récupérer le nom d'un pays à partir d'un code pays.

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tSOAP** et **tLogRow**.

2. Cliquez-droit sur le composant **tSOAP** afin d'ouvrir le menu contextuel. Sélectionnez **Row** > **Main** et cliquez sur le **tLogRow** pour relier les composants à l'aide d'un lien **Row** > **Main**.

3. Double-cliquez sur le **tSOAP** afin d'ouvrir sa vue **Basic settings** et définir ses propriétés.

4. Dans le champ **ENDPOINT**, saisissez ou collez l'URL du Service Web à utiliser, entre guillemets : *"http://www.webservicex.net/country.asmx"*.

5. Dans le champ **SOAP Action**, saisissez ou collez l'URL de l'en-tête HTTP SOAPAction permettant de déterminer que vous souhaitez récupérer les informations concernant le nom du pays : *http://www.webserviceX.NET/GetCountryByCountryCode*.

Vous pouvez voir cette adresse en regardant le WSDL du Service Web que vous appelez. Pour le Service Web de cet exemple, dans votre navigateur, saisissez `?wsdl` à la fin de l'URL du Service Web utilisé dans le champ **ENDPOINT**, ouvrez la page Web correspondante, puis regardez l'action SOAPAction définie sous le nœud Operation :

```
<wsdl:operation name="GetCountryByCountryCode">
<soap:operation style="document"  soapAction="http://www.webserviceX.NET/
GetCountryByCountryCode"/>
```

6. Dans le champ **SOAP version**, sélectionnez la version du système SOAP utilisé. Dans ce scénario, la version est *SOAP 1.1*.

7. Dans le champ **SOAP message**, saisissez le message, au format XML, utilisé pour récupérer les informations de pays du Service Web invoqué. Dans cet exemple, *IS* est utilisé comme code pays, le message est donc :

```
"<soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\"
 xmlns:web=\"http://www.webserviceX.NET\">
   <soapenv:Header/>
   <soapenv:Body>
      <web:GetCountryByCountryCode>
         <!--Optional:-->
         <web:CountryCode>IS</web:CountryCode>
      </web:GetCountryByCountryCode>
   </soapenv:Body>
</soapenv:Envelope>"
```

8. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

Le nom du pays est récupéré et affiché dans la console de la vue **Run** à partir du code pays *IS*.

```
[statistics] connecting to socket on port 3879
[statistics] connected
|<GetCountryByCountryCodeResponse
xmlns="http://www.webserviceX.NET"><GetCountryByCountryCodeResult>&lt;NewDataSet&gt;
  &lt;Table&gt;
    &lt;countrycode&gt;is&lt;/countrycode&gt;
    &lt;name&gt;Iceland&lt;/name&gt;
  &lt;/Table&gt;
  &lt;Table&gt;
    &lt;countrycode&gt;is&lt;/countrycode&gt;
    &lt;name&gt;Iceland&lt;/name&gt;
  &lt;/Table&gt;
&lt;/NewDataSet&gt;</GetCountryByCountryCodeResult></GetCountryByCountryCodeResponse>|
[statistics] disconnected
```

Scénario 2 : Utiliser un message SOAP depuis un fichier XML pour obtenir le nom d'un pays et le sauvegarder dans un fichier XML

Ce scénario décrit un Job à trois composants utilisant un message SOAP d'un fichier d'entrée XML afin d'invoquer un service Web. Il permet également récupérer le nom d'un pays correspondant à un code pays donné, dans cet exemple, *IR*, puis d'écrire la réponse dans un fichier XML.

Déposer et relier les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tFileInputXML**, un **tSOAP** et un **tFileOutputXML**.

2. Connectez les composants à l'aide de liens **Main** > **Row**.

Configurer le composant d'entrée

1. Double-cliquez sur le composant **tFileInputXML** pour ouvrir sa vue **Basic settings** et configurer ses propriétés.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir la boîte de dialogue **[Schema]**.

3. Cliquez sur le bouton **[+]** pour ajouter une colonne et nommez-la *getCountryName* dans ce scénario. Sélectionnez le type **Document** dans la liste **Type**, puis cliquez sur **OK** pour fermer la boîte de dialogue.

4. Dans le champ **File name/Stream**, saisissez le chemin d'accès au fichier d'entrée XML contenant le message SOAP à utiliser, ou parcourez votre système jusqu'à ce fichier en cliquant sur le bouton **[...]**.

   Le fichier d'entrée contient le message SOAP suivant. Vous pouvez constater que le code pays donné est *IR*.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:web="http://www.webserviceX.NET">
    <soapenv:Header/>
    <soapenv:Body>
        <web:GetCountryByCountryCode>
            <web:CountryCode>IR</web:CountryCode>
        </web:GetCountryByCountryCode>
    </soapenv:Body>
</soapenv:Envelope>
```

5. Dans le champ **Loop XPath query**, saisissez `"/"` afin de définir l'élément racine comme nœud de boucle de la structure du fichier d'entrée.

6. Dans la table **Mapping**, renseignez la colonne **XPath query** en saisissant `" . "` pour extraire toutes les données du nœud contexte de la source. Cochez la case **Get Nodes** pour construire un flux de données de type **Document**.

Configurer le service Web à l'aide du tSOAP

1. Double-cliquez sur le composant **tSOAP** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **ENDPOINT**, saisissez ou collez l'URL du service Web à utiliser, entre guillemets : *"http://www.webservicex.net/country.asmx"*.

3. Dans le champ **SOAP Action**, saisissez ou collez l'URL de l'en-tête HTTP SOAPAction indiquant que vous souhaitez récupérer les informations du pays : *http://www.webserviceX.NET/GetCountryByCountryCode*.

4. Cochez la case **Use a message from the input schema** et sélectionnez une colonne de type **Document** dans la liste **SOAP Message** afin de lire le message SOAP du fichier d'entrée et de l'envoyer au service Web. Dans cet exemple, le schéma d'entrée possède une seule colonne, *getCountryName*.

5. Cochez la case **Output in Document** pour écrire en sortie le message de réponse au format XML.

Configurer le composant de sortie

1. Double-cliquez sur le composant **tFileOutputXML** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **File Name**, saisissez le chemin d'accès au fichier XML de sortie.

3. Cochez la case **Incoming record is a document** afin de récupérer le flux de données entrant en tant que document XML. Notez qu'une liste **Column list** apparaît, vous permettant de sélectionner une colonne de laquelle récupérer les données. Dans cet exemple, le schéma contient une seule colonne.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur la touche **F6** ou cliquez sur le bouton **Run** dans l'onglet **Run** pour exécuter le Job.

   Les informations concernant le pays correspondant au code pays *IR* sont retournées et sauvegardées dans le fichier XML défini.

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetCountryByCountryCodeResponse xmlns="http://www.webserviceX.NET">
      <GetCountryByCountryCodeResult>&lt;NewDataSet&gt; &lt;Table&gt;
        &lt;countrycode&gt;ir&lt;/countrycode&gt; &lt;name&gt;Iran&lt;/name&gt; &lt;/Table&gt;
        &lt;Table&gt; &lt;countrycode&gt;ir&lt;/countrycode&gt; &lt;name&gt;Iran&lt;/name&gt;
        &lt;/Table&gt; &lt;/NewDataSet&gt;</GetCountryByCountryCodeResult>
    </GetCountryByCountryCodeResponse>
  </soap:Body>
</soap:Envelope>
```

tSVNLogInput

Propriétés du tSVNLogInput

Famille de composant	Internet	
Fonction	Le composant **tSVNLogInput** récupère les informations d'une ou plusieurs révisions d'un référentiel SVN.	
Objectif	Ce composant permet de récupérer les informations d'une révision spécifiée ou d'un ensemble de révisions d'un référentiel SVN.	
Basic settings	*URL*	Saisissez l'URL du référentiel SVN auquel accéder.
	Need authentication?	Cochez cette case si l'authentification est requise pour accéder au référentiel SVN.
	Username et *Password*	Saisissez le nom et le mot de passe pour accéder au référentiel SVN si une authentification est nécessaire. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Ces deux champs sont disponibles uniquement lorsque la case **Need authentication?** est cochée.
	From revision	Spécifiez un numéro de révision de début pour l'ensemble de révisions.
	Only this revision	Cochez cette case si vous souhaitez récupérer les informations de la seule révision spécifiée dans le champ **From revision**.
	To revision	Spécifiez un numéro de révision de fin pour l'ensemble de révisions. Saisissez *'HEAD'* pour la dernière révision. Le champ disparaît lorsque la case **Only this revision** est cochée.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit les champs à traiter et à passer au composant suivant. Le schéma de ce composant est en lecture seule. Il décrit les propriétés de la révision. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour voir le schéma prédéfini contenant les champs suivants : • **revision** : numéro de révision du message de log. • **author** : auteur du message de commit. • **date** : date du message de commit. • **message** : contenu du message. • **nb_file_added** : nombre de fichiers ajoutés. • **nb_file_modified** : nombre de fichiers modifiés. • **nb_file_deleted** : nombre de fichiers supprimés.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job et de chaque composant.
Variables globales	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé comme composant d'entrée dans un Job et nécessite un lien de sortie.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate **Trigger** : On Subjob Ok, On Subjob Error, Run if, On Component Ok, On Component Error Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Iterate **Trigger** : On Subjob Ok, On Subjob Error, Run if, On Component Ok, On Component Error Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.
Limitation	N/A

Scénario : Récupérer un message de log depuis un référentiel SVN

Ce scénario décrit un Job de deux composants récupérant les informations d'une révision spécifiée depuis un référentiel SVN et affiche les informations dans la console.

1. Créez un nouveau Job et ajoutez les deux composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette** : un **tSVNLogInput** et un **tLogRow**.

2. Reliez le **tSVNLogInput** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

3. Double-cliquez sur le **tSVNLogInput** pour ouvrir sa vue **Basic settings**.

4. Dans le champ **URL**, saisissez l'URL du référentiel SVN auquel accéder. Dans cet exemple, saisissez *https://talend.com/svndoc/*. Notez que les informations d'authentification fournies ici sont pour la démonstration.

5. Cochez la case **Need authentication?** et renseignez les champs **Username** et **Password** avec vos informations d'authentification.

6. Spécifiez une révision ou un ensemble de révisions. Dans cet exemple, récupérez les informations concernant la révision 17090, saisissez donc *17090* dans le champ **From revision** et cochez la case **Only this revision**.

7. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

8. Définissez la manière dont vous souhaitez afficher les données de sortie dans la console. Dans cet exemple, saisissez *"\r\n"* dans le champ **Field Separator** pour afficher chaque champ d'information dans une nouvelle ligne.

9. Cochez la case **Print schema column name in front of each value** pour afficher le nom de chaque champ concernant les informations de révision au début de chaque ligne.

10. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job puis sur **F6** pour l'exécuter.

```
[statistics] connecting to socket on port 3665
[statistics] connected
revision: 17090
author: lli
date: Fri Oct 24 16:03:03 CST 2014
message: Updated the Properties table for tOracleCDC
Added a new component tOracleCDCOutput with a scenario
Updated the chapter file ch-components-databases-traditional.xml

DOCT-3519, arch used, keywords added, changes tracked
nb_file_added: 16
nb_file_modified: 2
nb_file_deleted: 0
[statistics] disconnected
```

Comme montré dans la capture d'écran, les informations de la révision spécifiée sont affichées dans la console.

tWebService

Ce composant requiert l'utilisation d'un JDK Oracle.

Propriétés du tWebService

Famille de composant	Internet	
Fonction	Ce composant appelle la méthode du service Web invoqué et retourne la classe définie en fonction des paramètres donnés.	
Objectif	Le **tWebService** invoque une méthode via un service Web afin de récupérer les valeurs des paramètres sélectionnés dans l'éditeur.	
Basic settings	*Property type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Service configuration	Cliquez sur le bouton **[...]** pour ouvrir l'éditeur du composant. Dans l'éditeur, vous pouvez : -sélectionner le service Web que vous souhaitez utiliser, -configurer les paramètres d'entrée du service Web, -configurer les paramètres de sortie du service Web permettant de récupérer les données à transmettre en sortie.
	Mapping links display as	**Auto** : par défaut, les liens faisant la correspondance entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbes. **Curves** : les liens faisant la correspondance entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbes. **Lines** : les liens faisant la correspondance entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de lignes droites. Cette dernière option améliore légèrement les performances.
	Connection Time out	Saisissez en secondes la durée de connexion au service Web.
	Receive Time out	Saisissez en secondes le temps de réponse du serveur.
	Input schema	Un schéma est une description de lignes, il définit le nombre de champ qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Ce champ est utilisé pour traiter le schéma d'entrée. Les options pour ce schéma sont : - **Built-in** : Propriétés utilisées ponctuellement. - **Repository** : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.

	Edit Schema	Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
	Sync columns	Ce bouton est disponible lorsqu'un lien d'entrée est créé. Cliquez sur ce bouton pour récupérer le schéma du composant précédent.
	Output schema	Ce champ est utilisé pour traiter le schéma de sortie. Le schéma est soit built-in, soit distant dans le Repository. Il est configuré comme le schéma d'entrée. *Le schéma d'entrée n'est pas nécessairement identique à celui du schéma de sortie.* Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
	Use NTLM	Cochez cette case si vous utilisez un protocole d'authentification NTLM. **Domain** : Nom de domaine du client, **Host** : Adresse IP du client.
	Need authentication	Cochez la case d'authentification et renseignez le nom de l'utilisateur dans le champ **Username** et son mot de passe dans le champ **Password**, si cela est nécessaire pour accéder au service. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use http proxy	Cochez cette case si vous vous connectez derrière un proxy et renseignez les informations correspondantes.
	Trust server with SSL	Cochez cette case pour authentifier le serveur auprès du client via un protocole SSL et renseignez les champs correspondants. **TrustStore file** : saisissez le chemin d'accès et le nom du fichier TrustStore contenant la liste des certificats approuvés par le client. **TrustStore password** : saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore.
	Die on error	Décochez cette case pour passer les lignes en erreur et terminer le traitement des lignes sans erreur.
Advanced settings	*Temporary folder (for wsdl2java)*	Définissez (ou parcourez votre répertoire jusqu'à) un dossier temporaire configuré où stocker les fichiers wsdl.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
	Enable parallel execution	Cocher cette case permet de traiter plusieurs flux de données simultanément. Cela permet de traiter les données plus rapidement. Notez que cette fonctionnalité dépend de la capacité de la base de données ou de l'application à gérer plusieurs insertions en parallèle ainsi que le nombre de processeurs utilisés. Dans le champ **Number of parallel executions**, vous pouvez : • Saisir le nombre d'exécutions parallèles désiré.

		• Appuyer sur **Ctrl+Espace** et sélectionner la variable de contexte dans la liste. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*. Notez que, lorsque l'exécution parallèle est activée, il n'est pas possible d'utiliser les variables globales pour récupérer les valeurs de retour dans un sous-job.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé comme composant intermédiaire ou comme composant de début de Job. Il requiert un composant de sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Un JDK est requis pour faire fonctionner ce composant.	

Scénario : Obtenir des noms de pays à l'aide d'un service Web

Ce scénario décrit un Job à trois composants utilisant un service Web pour récupérer le nom du pays correspondant à un code de pays donné.

Relier les composants

1. Déposez un **tFixedFlowInput**, un **tWebService** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.

2. Reliez les composants à l'aide de liens de type **Main Row**.

Configurer les données d'entrée

1. Double-cliquez sur le composant **tFixedFlowInput** afin d'afficher sa vue **Basic settings** et définir ses propriétés.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir le schéma du composant d'entrée.

3. Dans la boîte de dialogue, cliquez sur le bouton **[+]** pour ajouter une colonne au schéma.

4. Renommez cette colonne *CountryCode*.

5. Cliquez sur **OK** pour fermer la boîte de dialogue. La colonne *CountryCode* s'affiche dans la table **Values** de la vue **Basic settings** du composant.

6. Dans le tableau **Values**, cliquez sur la colonne **Value** et saisissez la valeur d'entrée dans la colonne *CountryCode*, *cn* dans cet exemple. Ce code de pays est passé au **tWebService** afin de récupérer le nom du pays correspondant.

Configurer le service Web

Sélectionnez le WSDL

1. Double-cliquez sur le composant **tWebService** pour ouvrir l'éditeur du composant. Vous pouvez également sélectionner le composant dans l'espace de modélisation graphique et, dans la vue **Basic settings**, cliquez sur le bouton **[...]** à côté du champ **Service configuration**.

2. Dans le champ **WSDL**, saisissez l'adresse WSDL du service Web que vous souhaitez utiliser ou cliquez sur le bouton **Browse...** pour sélectionner un WSDL stocké localement. Dans cet exemple, l'URL WSDL est *http://www.webservicex.net/ country.asmx?WSDL*.

3. Cliquez sur le bouton **Refresh** pour récupérer la description du WSDL dans les champs suivants.

4. Dans la liste **Port Name**, sélectionnez le port que vous souhaitez utiliser. Ici, sélectionnez le port *coutrySoap*.

5. Dans la liste **Operation**, sélectionnez le service que vous souhaitez utiliser. Ici, sélectionnez le service *GetCountryByCountryCode(parameters):string*.

Configurer le mapping d'entrée

1. Cliquez sur **Next** pour passer à l'étape suivante.

2. Dans le panneau droit de la vue **Input mapping**, sélectionnez l'élément **[+] parameters** de la zone de droite, en cliquant sur le bouton **[+]**, et en sélectionnant les paramètres souhaités dans la boîte de dialogue **[Parameter Tree]**.

   Le service Web utilisé dans ce scénario ne comporte qu'un seul paramètre d'entrée : *CountryCode*.

   S'il est disponible, utilisez le bouton **Auto map!** situé en bas à gauche de la fenêtre, il permet d'effectuer l'opération de mapping automatiquement.

   Vous devez créer une connexion entre le schéma d'entrée et les paramètres d'entrée du Service Web défini.

3. Dans la liste **Column**, sélectionnez la colonne du schéma d'entrée que vous souhaitez mettre en correspondance avec le paramètre de sortie du service Web et glissez-la dans le paramètre correspondant dans la zone de droite.

Configurer le mapping de sortie

1. Cliquez sur **Next** pour ouvrir une nouvelle vue dans l'éditeur.

Dans la liste **Element** à gauche de l'éditeur, le paramètre de sortie du service Web apparaît automatiquement. Vous avez la possibilité d'ajouter d'autres paramètres en sélectionnant l'élément **[+] parameters** de la liste **Element** à gauche de l'éditeur, en cliquant sur le bouton **[+]**, et en sélectionnant les paramètres souhaités dans la boîte de dialogue **[Parameter Tree]**.

Le service Web utilisé dans ce scénario ne comporte qu'un seul paramètre de sortie : *GetCountryByCountryCodeResult*.

2. Dans la zone de droite, cliquez sur le bouton **[...]** du champ **Edit Schema** pour définir le schéma de sortie.

3. Dans la zone **Output** à droite de la boîte de dialogue, cliquez sur le bouton **[+]** pour ajouter une colonne au schéma de sortie.

4. Nommez cette colonne *Result*. Elle représente le nom du pays.

5. Cliquez sur **OK** pour valider les changements et fermer la boîte de dialogue de définition de schéma.

   Vous devez à présent créer une connexion entre le paramètre de sortie du service Web défini et le schéma du composant de sortie.

6. Dans la liste **Element** à droite de l'éditeur, sélectionnez l'élément *parameters.GetCountryByCountryCodeResult* à gauche et glissez-le dans le champ correspondant à la colonne *Result* dans la zone de droite.

S'il est disponible, utilisez le bouton **Auto map!** situé en bas à gauche de la fenêtre, il permet d'effectuer l'opération de mapping automatiquement.

7. Cliquez sur **OK** pour valider les changements et fermer l'éditeur.

8. Cliquez sur le bouton **[...]** afin d'ouvrir la boîte de dialogue **[Schema]** ajoutez une colonne au schéma de sortie, *Result* dans cet exemple. Cliquez sur **OK** afin de valider les changements et fermer la boîte de dialogue.

9. Double-cliquez sur le **tLogRow** afin d'ouvrir sa vue **Basic settings** et cliquez sur **Sync columns** afin de récupérer le schéma du composant précédent.

Exécuter le Job

- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

Le nom du pays correspondant au code de pays défini est retourné par le service Web.

tWebServiceInput

Propriétés du tWebServiceInput

Famille de composant	Internet	
Fonction	Ce composant appelle la méthode du service Web invoqué et retourne la classe définie en fonction des paramètres donnés.	
Objectif	Le **tWebServiceInput** invoque une méthode via un service Web dont l'objectif est celui décrit dans les propriétés. Pour traiter des données complexes hiérarchiques, vous devez utiliser les fonctions avancées du **tWebServiceInput** en codant directement du Java dans le champ **Code** de la vue **Advanced settings**.	
Basic settings	*Property type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
		Cliquez sur cette icône pour ouvrir l'assistant de création de schéma WSDL et enregistrer les paramètres de connexion WSDL que vous avez définis dans la vue **Basic settings** du composant dans le Repository. Pour plus d'informations sur la configuration et le stockage des paramètres de connexion WSDL, consultez le *Guide utilisateur du Studio Talend*.
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champ qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

	WSDL	Description des bindings et de la configuration du service Web.
	Need authentication / Username et Password	Cochez cette case d'authentification et : - Saisissez un identifiant, ainsi qu'un mot de passe, dans les champs correspondants, si nécessaire pour accéder au service. Sinon, - Cochez la case **Windows authentication** et saisissez le nom de domaine Windows dans le champ correspondant si cela est nécessaire pour accéder au service. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Use http proxy	Cochez cette case si vous vous connectez derrière un proxy et renseignez les informations correspondantes.
	Trust server with SSL	Cochez cette case pour authentifier le serveur auprès du client via un protocole SSL et renseignez les champs correspondants. **TrustStore file** : saisissez le chemin d'accès et le nom du fichier TrustStore contenant la liste des certificats approuvés par le client. **TrustStore password** : saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore.
	Time out (second)	Saisissez en secondes la durée de connexion au service Web.
	Method Name	Saisissez le nom exact de la méthode à invoquer. Le nom de la méthode DOIT correspondre à la méthode décrite dans le service Web. Le nom de la méthode est sensible à la casse.
	Parameters	Saisissez les paramètres attendus et les valeurs à retourner. Assurez-vous que les paramètres saisis correspondent parfaitement aux noms et à la casse des paramètres décrits dans la méthode.
Advanced settings	*Advanced Use*	Cochez cette case pour afficher les champs suivants permettant d'utiliser les fonctions avancées du **tWebServiceInput** : **WSDL2Java** : cliquez sur le bouton **[...]** pour générer des routines contenant les codes Java nécessaires à la connexion et à l'interrogation du service Web. **Code** : saisissez le code permettant de se connecter et d'interroger le service Web en vous basant sur les deux modèles proposés en utilisant les routines générées automatiquement. **Match Brackets** : sélectionnez le nombre d'accolades à utiliser pour fermer la boucle `for` en fonction du nombre d'accolades ouvertes.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant est généralement utilisé comme composant de début. Il requiert un composant de sortie.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Obtenir des noms de pays à l'aide d'un Webservice

Ce scénario décrit un Job constitué de deux composants dont le but est d'utiliser une méthode de service Web afin d'obtenir le nom du pays correspondant à un code de pays donné et d'afficher le résultat en sortie dans la console **Run**.

1. Déposez un **tWebServiceInput** et un **tLogRow** depuis la **Palette** dans l'espace de modélisation graphique.

2. Reliez les deux composants à l'aide d'un lien **Row** > **Main**.

3. Double-cliquez sur le **tWebServiceInput** afin d'ouvrir sa vue **Basic settings**.

4. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir la boîte de dialogue **[Schema]** et de définir le schéma pour la réponse du service Web.

   Dans cet exemple, le schéma est composé d'une seule colonne, *Result*.

5. Dans le champ **WSDL**, spécifiez l'URL du service Web utilisé, *http://www.webservicex.net/country.asmx?WSDL* dans cet exemple.

6. Si vous vous connectez derrière un proxy, cochez la case **Use http proxy** et renseignez les champs concernant l'hôte, l'utilisateur, le port et le mot de passe.

7. Dans le champ **Method name**, saisissez le nom de la méthode à utiliser. Notez que le nom de la méthode est sensible à la casse.

   Dans cet exemple, le méthode *GetCountryByCountryCode* est utilisée.

8. Dans la zone **Parameters**, cliquez sur le bouton **[+]** pour ajouter une ligne au tableau, puis saisissez un code de pays, *fr* dans cet exemple.

9. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

L'information sur le pays est retournée depuis le service Web et affichée dans la console **Run**.

tXMLRPCInput

Propriétés du tXMLRPCInput

Famille de composant	Internet	
Fonction	Ce composant appelle la méthode du service RPC invoqué et retourne la classe définie en fonction des paramètres donnés.	
Objectif	Le composant **tXMLRPCInput** invoque une méthode via un service Web dont l'objectif est celui décrit dans les propriétés.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Dans le contexte du RPC, le schéma correspond aux paramètres de sortie. Si deux paramètres doivent être retournés, alors le schéma comportera deux colonnes.
	Server URL	Adresse URL du service RPC à accéder
	Need authentication / Username et Password	Cochez la case d'authentification et renseignez le nom de l'utilisateur (username) et son mot de passe (password), si cela est nécessaire pour accéder au service.
	Method Name	Saisissez le nom exact de la méthode à invoquer. Le nom de la méthode DOIT correspondre parfaitement à la méthode décrite dans le service RPC. Le nom de la méthode est sensible à la casse.
	Return class	Sélectionnez le type de données à retourner par la méthode. Assurez-vous qu'il correspond parfaitement à celui défini dans la méthode.
	Parameters	Saisissez les paramètres attendus en entrée par la méthode.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Chercher le nom d'un Etat via une méthode XMLRPC

Ce scénario décrit un Job à deux composants utilisant une méthode RPC et affichant le résultat dans la console.

- Cliquez et déposez les composants **tXMLRPCInput** et **tLogRow**.
- Dans l'onglet **Basic settings**, paramétrez les propriétés du composant **tXMLRPCInput**.

- Dans la liste déroulante **Schema type**, sélectionnez l'option **Built-in**.
- Configurez un schéma contenant une unique colonne puisque seul un paramètre est attendu en sortie : *StateName*.

- Dans le champ **Server url**, renseignez l'adresse URL du serveur. Pour cet exemple, utilisez : *http://phpxmlrpc.sourceforge.net/server.php*
- Aucune information d'authentification n'est nécessaire pour cet exemple.
- Dans le champ **Method**, renseignez la méhode à appeler, ici : *examples.getStateName*
- Pour cet exemple, la classe retournée n'est pas obligatoire. Ainsi, dans le champ **return class**, laissez les paramètres par défaut.
- Puis, dans le tableau **Parameters**, configurez les paramètres nécessaires à la méthode appelée. La colonne **Name** n'est pas utilisée dans le code mais la valeur doit correspondre à la syntaxe attendue par la méthode. Dans cet exemple, saisissez *State Nr* dans le champ **Name** et dans le champ **Value**, choisissez un numéro au hasard, ici : *42*.
- La classe n'a pas beaucoup d'impact sur cette méthode démo mais pourrait en avoir sur une autre méthode. Ainsi, laissez les paramètres par défaut.
- Dans la vue **Component** du composant **tLogRow**, cochez la case **Print schema column name in front of each value**.
- Enregistrez le Job puis exécutez-le, en appuyant sur **F6**.

```
Starting job xmlrpc at 16:25 21/09/2007.
StateName: South Dakota
Job xmlrpc ended at 16:25 21/09/2007. [exit code=0]
```

Le résultat obtenu en sortie en utilisant la méthode RPC `GetStateName` et correspondant au quarante-deuxième Etat des Etats-Unis est donc *South Dakota*.

Composants Logs & Errors

Ce chapitre passe en revue les principaux composants de la famille **Logs & Errors** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **Logs & Errors** regroupe les composants dédiés à la gestion des données de log et des erreurs des Jobs.

tAssert

Propriétés du tAssert

Le composant **tAssert** est étroitement lié au **tAssertCatcher** pour tester une exécution de Job. Il génère une assertion booléenne à partir d'une instruction conditionnelle, dans le but d'alimenter le statut d'exécution présenté par le **tAssertCatcher**.

Famille de composant	Logs & Errors	
Fonction	Envoie le message de statut du Job au **tAssertCatcher**.	
Objectif	Génère l'assertion booléenne relative à l'analyse du Job. Le statut peut être : **Ok** : l'exécution du Job a réussi. **Fail** : l'exécution a échoué. Les résultats du Job testé ne correspondent pas à ce qui était attendu, ou une erreur est survenue lors de l'exécution.	
Basic settings	*Description*	Saisissez votre message descriptif afin d'identifier l'assertion du **tAssert**.
	Expression	Saisissez l'instruction conditionnelle sur laquelle baser votre assertion.
Utilisation	Ce composant suit l'action à laquelle est liée directement la condition assertive. Il peut être utilisé comme composant intermédiaire ou final du Job principal, ou bien en tant que composant de début, intermédiaire ou de fin du Job secondaire.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Limitation	L'évaluation du **tAssert** est capturée par le **tAssertCatcher**.	

Scénario 1: Obtenir le statut des commandes d'un produit (quotidiennement) par rapport à un nombre fixe

Ce scénario permet d'insérer dans une table de base de données les informations liées à des commandes, ainsi que d'obtenir chaque jour le statut de ces commandes en utilisant le composant **tAssert** pour comparer les commandes à un nombre fixe et **tAssertCatcher** pour indiquer les résultats. Ici, **Ok** est retourné lorsque le nombre de commandes est supérieur à 20 et **Failed** est retourné s'il est inférieur à 20.

En pratique, ce Job peut être planifié pour s'exécuter tous les jours et fournir un rapport quotidien des commandes, et vous pouvez aussi remplacer les composants **tFixedFlowInput** et **tLogRow** par des composants d'entrée et de sortie dans les familles **Database** et **File**.

Déposer et relier les composants

1. Déposez les composants **tFixedFlowInput**, **tMysqlOutput**, **tAssert**, **tAssertCatcher**, et **tLogRow** de la **Palette** dans l'espace de modélisation graphique..

2. Renommez **tFixedFlowInput** en **orders**, **tAssert** en **orders >=20**, **tAssertCatcher** en **catch comparison result** et **tLogRow** en **ok or failed**.

3. Reliez le composant **tFixedFlowInput** à **tMysqlOutput** à l'aide d'un lien **Row** > **Main**.

4. Reliez le composant **tFixedFlowInput** à **tAssert** à l'aide d'un lien **Trigger** > **On Subjob Ok**.

5. Reliez le composant **tAssertCatcher** à **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

Sélectionnez **Use Inline Content (delimited file)** dans la zone **Mode**.

dans le champ **Content**, saisissez les données à passer en base Mysql, par exemple:

```
AS2152;Washingto Berry Juice;2013-02-19 11:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 13:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 14:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washingto Berry Juice;2013-02-19 12:14:15;3.6
```

Notez que ces commandes, inférieures à 20, sont données à titre d'exemple du fonctionnement du **tAssert**.

2. Cliquez sur le bouton **Edit schema** pour ouvrir l'éditeur de schéma.

3. Cliquez sur le bouton **[+]** pour ajouter quatre colonnes, nommées respectivement *product_id*, *product_name*, *date* et *price*, de type **String**, **Date**, **Float**.

   Cliquez sur **OK** pour valider la configuration et fermer l'éditeur.

4. Double-cliquez sur le **tMysqlOutput** pour ouvrir sa vue **Basic settings**.

5. Dans les champs **Host**, **Port**, **Database**, **Username** et **Password**, saisissez les identifiants et les informations de connexion à la base.

6. Dans le champ **Table**, saisissez le nom de la table, *order* par exemple.

7. Dans la liste **Action on table**, sélectionnez l'option **Drop table if exists and create**.

8. Dans la liste **Action on data**, sélectionnez l'option **Insert**.

9. Double-cliquez sur le **tAssert** pour ouvrir sa vue **Basic settings**.

10. Dans le champ **description**, saisissez les informations descriptives de l'objectif du **tAssert** dans ce scénario.

11. Dans le champ **expression**, saisissez l'expression permettant de comparer les données au nombre fixe:

```
((Integer)globalMap.get("tMysqlOutput_1_NB_LINE_INSERTED"))>=20
```

12. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

13. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour afficher le résultat sous forme de tableau.

Exécuter le Job

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** pour l'exécuter.

```
+-------------------+------+-------+-----------------+--------+---------+------+--------------------+-------------------------------+
|                                                                ok or failed                                                        |
=-------------------+------+-------+-----------------+--------+---------+------+--------------------+-------------------------------=
|moment             |pid   |project|job              |language|origin   |status|substatus           |description                    |
=-------------------+------+-------+-----------------+--------+---------+------+--------------------+-------------------------------=
|2013-04-10 17:05:35|skTa6D|LLL    |DemoJobFortAssert|java    |tAssert_1|Failed|Test logically failed|The number of orders geater than 20?|
+-------------------+------+-------+-----------------+--------+---------+------+--------------------+-------------------------------+
```

Comme montré dans la capture d'écran, le statut des commandes indique *Failed* puisque le nombre de celles-ci est inférieur à 20.

Scénario 2 : Paramétrer une condition assertive pour l'exécution d'un Job

Ce scénario décrit comment paramétrer une condition assertive dans le composant **tAssert** afin d'évaluer si l'exécution du Job a réussi ou non. De plus, vous pouvez également trouver comment deux résultats différents d'évaluation s'affichent et comment les lire. Le scénario utilise un **tAssert**, mais aussi :

- un **tFileInputDelimited** et un **tFileOutputDelimited**. Les deux composants constituent le Job principal, duquel le statut d'exécution est évalué. Pour plus d'informations sur ces deux composants, consultez *tFileInputDelimited* et *tFileOutputDelimited*.
- un **tFileCompare**. Il réalise la comparaison entre le fichier de sortie du Job principal et un fichier de référence standard. Le résultat comparatif est évalué par le **tAssert** par rapport à la condition assertive configurée dans ses paramètres. Pour plus d'informations à propos du **tFileCompare**, consultez *tFileCompare*.
- un **tAssertCatcher**. Il capture l'évaluation générée par le **tAssert**. Pour plus d'informations à propos du composant **tAssertCatcher**, consultez *tAssertCatcher*.
- un **tLogRow**. Il vous permet de lire l'évaluation capturée. Pour plus d'informations à propos du **tLogRow**, consultez *tLogRow*.

Afin de créer le Job principal, procédez comme suit :

- Préparez un fichier délimité *.csv* comme fichier source lu par votre Job principal.
- Éditez deux lignes dans ce fichier. Les modifications n'ont pas d'importance, vous pouvez uniquement simplifier le contenu.
- Nommez-le *source.csv*.
- Dans le *Studio Talend*, créez un nouveau Job, que vous appellerez *JobAssertion*.
- Placez un **tFileInputDelimited** et un **tFileOutputDelimited** dans l'espace de modélisation graphique.
- Connectez- les à l'aide d'un lien **Row Main** afin de créer le Job principal.

- Double-cliquez sur le **tFileInputDelimited** afin d'ouvrir sa vue **Component**.
- Dans le champ **File Name** de la vue **Component**, renseignez le chemin d'accès à votre fichier *source.csv*.

- Sélectionnez dans la liste **Property Type** le mode **Built-In** et cliquez sur le bouton **[...]** de l'option **Edit schema** afin de définir les données à passer au **tFileOutputDelimited**. Dans ce scénario, sélectionnez les données présentées dans le fichier *source.csv* que vous avez créé.

Pour plus d'informations à propos des types de schéma, consultez le *Guide utilisateur du Studio Talend*.

- Définissez les autres paramètres dans les champs correspondants, selon le fichier *source.csv* que vous avez créé.
- Double-cliquez sur le **tFileOutputDelimited** afin d'ouvrir sa vue **Component**.
- Dans le champ **File Name** de la vue **Component**, renseignez le chemin d'accès à votre fichier de sortie, et laissez les autres champs comme définis par défaut.

- Appuyez sur **F6** afin d'exécuter le Job principal. Il lit votre fichier *source.csv*, passe les données au **tFileOutputDelimited** et écrit un fichier délimité de sortie, *out.csv*.

Continuez à modifier votre Job, afin de voir comment le **tAssert** évalue le statut d'exécution du Job principal.

- Renommez le fichier *out.csv* en *reference.csv*. Ce fichier est le résultat attendu de l'écriture du Job principal.
- Placez un **tFileCompare**, un **tAssert** et un **tLogRow** dans l'espace de modélisation graphique.
- Connectez-les avec un lien **Row Main**.
- Connectez le **tFileInputDelimited** au **tFileCompare** avec un lien **OnSubjobOk**.

- Double-cliquez sur le **tFileCompare** afin d'ouvrir sa vue **Component**.
- Dans la vue **Component**, renseignez les chemins d'accès dans les champs **File to compare** et **Reference file** correspondants, et laissez les autres champs comme définis par défaut.

Pour plus d'informations à propos du composant **tFileCompare**, consultez *tFileCompare*.

- Cliquez ensuite sur le **tAssert**, puis cliquez sur l'onglet **Component**, au bas de l'espace de modélisation graphique.

- Dans la vue **Component**, éditez la condition `row2.differ==0`, dans le champ **expression** ainsi que la **description** de l'assertion.

Dans le champ **expression**, `row2` constitue le flux de données transmis du **tFileCompare** au **tAssert**, `differ` est une colonne du schéma du **tFileCompare** qui affiche si les fichiers comparés sont identiques. 0 signifie qu'aucune différence n'a été détectée entre le fichier *out.csv* et le fichier *reference.csv* par le **tFileCompare**. Les fichiers comparés sont donc identiques, la condition assertive est remplie, le **tAssert** conclut que le Job principal a été exécuté avec succès. Sinon, il conclut qu'il a échoué.

La colonne *differ* fait partie du schéma, en lecture seule, du **tFileCompare**. Pour plus d'informations sur son schéma, consultez *tFileCompare*.

- Appuyez sur **F6** afin d'exécuter votre Job.

- Vérifiez le résultat retourné dans la vue **Run**.

```
Starting job JobAssertion at 14:00 29/01/2010.

[statistics] connecting to socket on port 3788
[statistics] connected
[job JobAssertion] Files are identical
.----------+----------------+-------------------+------------+--------------+------+---------
-----------------------------.
|                                                          tLogRow_1
|
|=---------+----------------+-------------------+------------+--------------+------+---------
-----------------------------=|
|file      |file_ref        |moment             |job         |component     |differ|message
|
|=---------+----------------+-------------------+------------+--------------+------+---------
-----------------------------=|
|C:/out.csv|C:/reference.csv|2010-01-29 14:00:17|JobAssertion|tFileCompare_1|0     |[job
JobAssertion] Files are identical|
'----------+----------------+-------------------+------------+--------------+------+---------
-----------------------------'
[statistics] disconnected
Job JobAssertion ended at 14:00 29/01/2010. [exit code=0]
```

La console affiche les résultats de la comparaison du **tFileCompare** : `Files are identical`. Mais l'évaluation du **tAssert** n'est visible nulle part.

Vous avez donc besoin du composant **tAssertCatcher** pour capturer l'évaluation.

- Placez un **tAssertCatcher** et un **tLogRow** dans l'espace de modélisation graphique.
- Connectez-les à l'aide d'un lien **Row Main**.

- Utilisez la configuration par défaut du **tAssertCatcher** dans sa vue **Component**.

- Appuyez sur **F6** afin d'exécuter votre Job.
- Vérifiez les résultats présentés dans la vue **Run**. Vous pouvez voir que les informations sur le statut du Job ont été ajoutées :

```
2010-01-29 15:37:33|fAvAzH|TASSERT|JobAssertion|java|tAssert_1|Ok|--| The output file
should be identical with the reference file.
```

```
Starting job JobAssertion at 19:33 01/02/2010.

[statistics] connecting to socket on port 3556
[statistics] connected
[job JobAssertion] Files are identical
2010-02-01
19:33:40|r9FqSQ|TASSERT|JobAssertion|java|tAssert_1|Ok|--|The
output file should be identical with the reference file.
```

La description du *JobAssertion* dans la console est organisée selon le schéma du **tAssertCatcher**, lequel comprend des colonnes correspondant au temps d'exécution, à l'ID traité, au nom du projet, au langage du code, l'origine de l'évaluation, le résultat de l'évaluation, les informations détaillées de l'évaluation, et le message descriptif de l'assertion. Pour plus d'informations à propos du schéma du **tAssertCatcher**, consultez *tAssertCatcher*.

La console indique que le statut d'exécution du Job *JobAssertion* est *Ok*. En plus de l'évaluation, vous pouvez toujours voir les informations descriptives du *JobAssertion*, parmi lesquelles la description que vous avez saisie dans les **Basic settings** du **tAssert**.

Effectuez ensuite quelques opérations pour que le Job principal ne puisse générer le fichier attendu. Pour ce faire, procédez comme suit :

- Effacez une ligne de votre fichier *reference.csv*.
- Appuyez sur **F6** pour exécuter votre Job à nouveau.
- Vérifiez les résultats retournés dans la vue **Run**.

```
2010-02-01 19:47:43|GeHJNO|TASSERT|JobAssertion|tAssert_1|Failed|Test
logically failed|The output file should be identical with the reference
file
```

```
Starting job JobAssertion at 19:47 01/02/2010.

[statistics] connecting to socket on port 4001
[statistics] connected
[job JobAssertion] Files differ
2010-02-01
19:47:43|GeHJNO|TASSERT|JobAssertion|java|tAssert_1|Failed|Test
logically failed|The output file should be identical with the
reference file.
```

La console indique que le statut d'exécution du Job principal est *Failed* (a échoué). L'explication détaillée suit peu après, *Test logically failed* (échec logique du test).

Vous pouvez avoir une idée du statut de votre Job : il a échoué lors de la génération du fichier attendu à cause d'un échec logique. Cet échec logique peut provenir d'une erreur logique lors de la création du Job.

Le statut et ses explications sont présentés respectivement dans les colonnes de statut et de sous-statut du schéma du **tAssertCatcher**. Pour plus d'informations à propos de cette colonne, consultez *tAssertCatcher*.

tAssertCatcher

Propriétés du tAssertCatcher

Famille de composant	Logs & Errors	
Fonction	A partir de son schéma prédéfini, le **tAssertCatcher** retrouve les informations des statuts d'exécution, des exécutions des Jobs et du **tAssert**.	
Objectif	Génère un flux de données consolidant les informations de statut d'une exécution de Job, et transfère les données dans les fichiers de sortie définis.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit les champs à traiter et à passer au composant suivant. Pour ce composant, le schéma est en lecture seule puisqu'il regroupe les informations de log standard, notamment :
		Moment : Date et heure de traitement.
		Pid : ID traités.
		Project : Projet auquel appartiennent les Jobs analysés.
		Job : Nom du Job analysé.
		Origin : Origine de l'assertion de statut. Elle peut provenir de différents composants **tAssert**.
		Status : Jugements retrouvés à partir du **tAssert**. Ils peuvent être : - **Ok** : si l'instruction conditionnelle du **tAssert** est évaluée comme vraie (*true*) à l'exécution. - **Failed** : si l'instruction conditionnelle du **tAssert** est évaluée comme fausse (*false)*, ou si des erreurs surviennent à l'exécution. Les résultats du Job testé ne correspondent pas à ce qui était attendu, ou une erreur est survenue lors de l'exécution.
		Substatus : Explications détaillées à propos des assertions de statut. Les explications peuvent être : - **Test logically failed** : le Job analysé ne produit pas le résultat attendu. - **Execution Error** : une erreur d'exécution est survenue lors de l'exécution.
		Description : La description textuelle saisie dans l'onglet **Basic settings** du **tAssert** (lorsque la case **Catch tAssert** est cochée) et/ou le message d'exception capturée (lorsque la case **Catch Java Exception** est cochée).
		Exception : l'objet de l'exception retournée par le Job, l'exception originale. Disponible lorsque la case **Get original exception** est cochée.
	Catch Java Exception	Cette case, une fois cochée, vous permet de capturer les erreurs d'exception Java et d'afficher le message dans la colonne **Description** (quand la case **Get original exception** n'est pas cochée) ou dans la colonne **Exception** (quand la case **Get original exception** est cochée).
	Get original exception	Cette case, une fois cochée, permet d'afficher l'objet de l'exception originale dans la colonne **Exception**. Disponible lorsque la case **Catch Java Exception** est cochée.
	Catch tAssert	Cette case vous permet de capturer les assertions du **tAssert**.

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant de début de Job secondaire qui retrouve les informations de statut d'exécution à partir de différentes sources. Il génère un flux de données afin de transférer les informations au composant suivant.
Limitation	Ce composant doit être utilisé avec le composant **tAssert**.

Scénario associé

Pour un scénario associé, consultez le scénario du **tAssert** :

- *Scénario 2 : Paramétrer une condition assertive pour l'exécution d'un Job*.

tChronometerStart

Propriétés du tChronometerStart

Famille de composant	Logs & Errors	
Fonction	Le composant **tChronometerStart** démarre l'opération de mesure du temps d'exécution d'un sous-Job.	
Objectif	Le **tChronometerStart** fonctionne comme un chronomètre : le composant commence à calculer soit le temps de traitement d'un ou plusieurs sous-Jobs du Job principal, soit le temps de traitement d'une partie d'un sous-Job.	
Global Variables	**STARTTIME** : heure de départ utilisée pour calculer le temps de traitement du (des) sous-job(s). Cette variable est une variable **Flow** et retourne un long. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Vous pouvez utiliser le **tChronometerStart** comme composant de début ou au milieu d'un Job. Il peut précéder une ou plusieurs tâches de traitement dans un sous-Job. Il peut également précéder un ou plusieurs sous-Jobs dans le Job principal.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Mesurer la durée de traitement d'un sous-Job ou d'une partie d'un sous-Job*.

tChronometerStop

Propriétés du tChronometerStop

Famille de composant	Logs & Errors	
Fonction	Le composant **tChronometerStop** mesure le temps d'exécution d'un sous-Job.	
Objectif	Le **tChronometerStop** fonctionne comme un chronomètre : Il affiche le temps d'exécution total et arrête le calcul du temps de traitement soit d'un ou de plusieurs sous-Jobs dans le Job principal, soit d'une partie d'un sous-Job.	
Basic settings	*Since options*	Cochez l'une des deux cases pour choisir le point de départ du processus de mesure : **Since the beginning** : arrête le calcul du temps dont la mesure a commencé au début d'un sous-Job. **Since a tChronometerStart** : arrête le calcul du temps dont la mesure a commencé à partir d'un composant **tChronometerStart** utilisé sur le flux de données du sous-Job.
	Display duration in console	Cochez cette case pour afficher les informations d'exécution du sous-Job sur la console.
	Display component name	Cochez cette case pour afficher le nom des composants sur la console.
	Caption	Entrez un texte, pour permettre d'identifier votre sous-Job par exemple.
	Display human readable duration	Cochez cette case pour afficher les informations d'exécution du sous-Job en unités de temps exploitables.
Global Variables	**STOPTIME** : heure de fin utilisée pour calculer le temps de traitement du (des) sous-job(s). Cette variable est une variable **Flow** et retourne un long. **DURATION** : temps de traitement du (des) sous-job(s). Cette variable est une variable **Flow** et retourne un long. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ne peut être utilisé comme composant de début.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Mesurer la durée de traitement d'un sous-Job ou d'une partie d'un sous-Job

Ce scénario est un sous-Job, qui, dans une séquence, effectue les opérations suivantes :

- génère 1 000 000 lignes de noms et de prénoms ;
- regroupe les noms et les prénoms correspondants ;
- stocke les données de sortie dans un fichier délimité ;
- mesure la durée d'ensemble du sous-Job ;
- mesure la durée de l'opération de remplacement des noms,
- affiche les informations relatives au temps de traitement dans la console de log **Run**.

Pour mesurer le temps d'exécution du sous-Job :

- A partir de la **Palette**, cliquez-déposez les composants suivants dans l'éditeur graphique : le **tRowGenerator**, le **tMap**, le **tFileOutputDelimited** et le **tChronometerStop**.
- Reliez les trois premiers composants à l'aide de liens de type **Main Row**.

Lorsque vous reliez le **tMap** au **tFileOutputDelimited**, vous serez invités à donner un nom à la table en sortie. Le nom utilisé dans cet exemple est "new_order".

- Reliez le **tFileOutputDelimited** au **tChronometerStop** à l'aide d'un lien de type **OnComponentOk**.
- Sélectionnez le **tRowGenerator** puis cliquez sur la vue **Component** pour paramétrer ses propriétés de base, définies par défaut, dans l'onglet **Basic settings**.

- Cliquez sur le bouton **[...]** du champ **Edit schema** pour définir le schéma du **tRowGenerator**. Cliquez deux fois sur le bouton **[+]** pour ajouter deux colonnes : *First_Name* et *Last_Name*. Cliquez sur **OK** afin de fermer la boîte de dialogue.

- Cliquez sur le bouton **[...]** du champ **RowGenerator Editor** pour ouvrir l'éditeur et définir les données à générer.

- Dans le champ **Number of Rows for RowGenerator** de l'éditeur **[RowGenerator Editor]**, précisez le nombre de lignes à générer puis cliquez sur **OK**. L'éditeur **[RowGenerator Editor]** se ferme.

- Une boîte de dialogue s'ouvre et vous demande si vous souhaitez propager les changements. Cliquez sur **Yes** (Oui).

- Double-cliquez sur le composant **tMap** pour ouvrir son éditeur. L'éditeur du **tMap** affiche le schéma du composant **tRowGenerator**.

- Dans l'onglet **Schema editor** de l'éditeur du **tMap**, cliquez sur le bouton **[+]** de la table de sortie pour ajouter deux lignes puis configurez-les.

- Dans l'éditeur du **tMap**, glissez la ligne *First_Name* de la table d'entrée dans la ligne *Last_Name* de la table de sortie et glissez la ligne *Last_Name* dans la ligne *First_Name* de la table de sortie.

- Cliquez sur **Apply** pour enregistrer les modifications.

- A nouveau, une boîte de dialogue s'ouvre et vous demande si vous souhaitez propager les changements. Cliquez sur **Yes**.

- Cliquez sur **OK** pour fermer l'éditeur.

- Sélectionnez le **tFileOutputDelimited** et cliquez sur la vue **Component** pour paramétrer ses propriétés de base dans l'onglet **Basic settings** :

- Sélectionnez le **tChronometerStop** et cliquez sur la vue **Component** pour paramétrer ses propriétés de base dans l'onglet **Basic settings**.

- Dans la zone **Since options,** cochez l'option **Since the beginning** pour mesurer la durée d'exécution du sous-Job entier.

- Cochez/décochez les autres cases en fonction de vos besoins. Dans ce scénario, l'objectif est d'afficher dans la console la durée d'exécution du sous-Job, précédée du nom du composant.
- Si nécessaire, saisissez du texte de le champ **Caption**.
- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job tChronometerStop at 11:40 05/03/2010.

[statistics] connecting to socket on port 3399
[statistics] connected
[ tChronometerStop_1 ]     2seconds   duration of the subjob
2968 milliseconds
[statistics] disconnected
Job tChronometerStop ended at 11:40 05/03/2010. [exit code=0]
```

Vous pouvez aussi mesurer la durée du sous-Job en plaçant un **tChronometerStop** en dessous du **tRowGenerator**, et en les connectant via un lien **OnSubJobOk**.

tDie

Propriétés du tDie

Les composants **tDie** et **tWarn** sont étroitement liés au composant **tLogCatcher**. Ils sont généralement utilisés ensemble afin que les données de log collectées par le tLogCatcher soient rassemblées et envoyées vers la sortie définie.

Famille de composant	Logs & Errors	
Fonction	Le composant **tDie** envoie une erreur et arrête le Job en cours d'exécution. Si vous souhaitez uniquement envoyer un message d'avertissement, consultez *tWarn*.	
Objectif	Ce composant déclenche le composant **tLogCatcher** afin de collecter des données de log exhaustives avant de tuer le Job.	
Basic settings	*Die message*	Saisissez le message qui sera affiché avant que le Job ne soit tué.
	Error code	Saisissez le code de l'erreur (un entier) si nécessaire.
	Priority	Sélectionnez le niveau de priorité.
Global Variables	**DIE_MESSAGES** : message d'arrêt. Cette variable est une variable **After** et retourne une chaîne de caractères. **DIE_CODE** : code d'erreur du message d'arrêt. Cette variable est une variable **After** et retourne un nombre entier. **DIE_PRIORITY** : niveau de priorité du message d'arrêt. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant ne peut être utilisé comme composant de début. Il est généralement utilisé avec un **tLogCatcher** afin de collecter des données de log.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour des exemples d'utilisation du composant **tDie**, consultez les scénarios du composant **tLogCatcher** :

- *Scénario 1 : Avertissement & log lors de la création d'entrées*.

- *Scénario 2 : Monitorer & arrêter un Job*.

tFlowMeter

Propriétés du tFlowMeter

Famille de composant	Logs & Errors	
Fonction	Le composant **tFlowMeter** compte le nombre de lignes traitées dans le flux spécifié.	
Objectif	Le nombre de lignes est ensuite récupéré par le composant **tFlowMeterCatcher** afin d'être ajouté dans les données de log.	
Basic settings	*Use input connection name as label*	Cochez cette case pour réutiliser le nom donné au flux d'entrée Main Row comme libellé des données loggées.
	Mode	Sélectionnez le type de valeur des données mesurées : **Absolute** : le nombre actuel de lignes est loggé. **Relative** : un pourcentage (%) du nombre de lignes est loggé. Lorsque cette option est sélectionnée, la liste **Connections List** vous permet de sélectionner une connexion de référence.
	Thresholds	Ajoutez un seuil pour observer les proportions en volumes mesurés. Vous pouvez décider que le flux doit être compris entre un nombre minimum et maximum de lignes, et si le flux est inférieur au seuil minimum, il y a un goulot d'étranglement.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant ne peut être utilisé comme composant de début puisqu'il nécessite un flux d'entrée pour fonctionner.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Si vous avez besoin d'utiliser les données de log, les statistiques et tout autre mesure des flux de données, consultez le *Guide utilisateur du Studio Talend*.

Scénario associé

Pour plus d'informations sur l'utilisation du **tFlowMeter**, consultez *Scénario : Mesurer le flux d'un Job*.

tFlowMeterCatcher

Propriétés du tFlowMeterCatcher

Famille de composant	Logs & Errors	
Fonction	Le composant **tFlowMeterCatcher** est basé sur le schéma défini. Il récupère les données du composant **tFlowMeter** et les transmet au composant de sortie.	
Objectif	Ce composant fonctionne comme une fonction de log déclenchée par l'utilisation du composant **tFlowMeter** dans le Job.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit les champs à traiter et à passer au composant suivant. Pour ce composant, le schéma est en lecture seule puisqu'il regroupe les informations de log standard, notamment :
		Moment : Date et heure du traitement
		Pid : Identifiant système du Job.
		Father_pid : Identifiant système du Job parent, le cas échéant. Sinon, le Pid est dupliqué.
		Root_pid : Identifiant système du Job racine, le cas échéant. Sinon, le Pid est dupliqué.
		System_pid : Identifiant système généré par le système.
		Project : Nom du projet auquel appartient le Job.
		Job : Nom du Job en cours.
		Job_repository_id : Identifiant généré par l'application.
		Job_version : Numéro de version du Job en cours.
		Context : Nom du contexte en cours.
		Origin : Nom du composant, le cas échéant.
		Label : Libellé de la connexion Main Row précédent le composant **tFlowMeter** dans le Job et dont la volumétrie sera analysée.
		Count : Nombre de lignes en cours de traitement.
		Reference : Nombre de lignes passant dans le lien **Reference**.
		Thresholds : cette fonction ne s'utilise que si le mode Relative a été sélectionné dans le composant **tFlowMeter**.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est le composant de début du Job secondaire qui se déclenche automatiquement à la fin du Job principal.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	L'utilisation de ce composant est liée à l'utilisation du composant **tFlowMeter**. Pour plus d'informations, consultez *tFlowMeterCatcher*.

Scénario : Mesurer le flux d'un Job

Le Job suivant permet de recueillir le nombre de lignes traitées. Les mesures sont prises deux fois, une fois après le composant d'entrée, c'est-à-dire avant qu'elles soient filtrées, et après leur filtrage, avant le composant de sortie.

- Cliquez et déposez les composants suivants : un **tMysqlInput**, deux **tFlowMeter**, un **tMap**, un **tLogRow**, un **tFlowMeterCatcher** et un **tFileOutputDelimited**.
- Connectez les composants du Job principal à l'aide de liens de type Row main et cliquez sur le libellé pour lui donner un nom plus parlant, par exemple *US_States* après le composant d'entrée et *filtered_states* après le **tMap**.
- Reliez aussi les composants **tFlowMeterCatcher** et **tFileOutputCSV** à l'aide d'une connexion de type **Row main** puisque les données sont transmises au composant de sortie.
- Dans la vue **Component** du composant **tMysqlInput**, sélectionnez l'option **Repository** dans le champ **Property Type**, si les propriétés de connexion à la base de données sont stockées dans une métadonnée du **Repository**. Sinon, sélectionnez l'option **Built-in** et paramétrez les propriétés et le schéma manuellement.

- Les 50 Etats des Etats-Unis sont enregistrés dans la table *states*. Pour que les 50 entrées soient sélectionnées, la requête à effectuer dans la base de données Mysql est la suivante :

```
select * from states.
```

- Dans l'onglet **Advanced settings**, sélectionnez l'encodage adéquat.
- Puis sélectionnez le composant suivant, le **tFlowMeter**, et paramétrez ses propriétés.

- Cochez la case **Use input connection name as label** pour réutiliser le nom de la connexion d'entrée dans le fichier log de sortie (**tFileOutputCSV**).
- Dans le champ **Mode**, sélectionnez l'option **Absolute** puisqu'il n'y a pas de flux de référence, et aucun seuil ne doit être défini dans le champ **Threshold** pour cet exemple.
- Puis lancez l'éditeur du **tMap** pour paramétrer les propriétés de filtrage.
- Pour cet exemple, glissez et déposez les colonnes ID et State de la zone Input (à gauche) du **tMap** vers la zone Output (à droite). Aucune variable n'est utilisée pour cet exemple.

- Dans la table du flux de sortie (ici, la table *filtered_states*), cliquez sur la flèche dotée d'un (+) pour ajouter un ligne de filtre.
- Glissez la colonne *State* de la zone Input (*row2*) vers le champ filtre et saisissez un filtre permettant de n'obtenir que les Etats commençant par la lettre *M*. L'expression finale ressemble à : `row2.State.startsWith("M")`
- Cliquez sur **OK** pour valider les paramètres.
- Puis sélectionnez le deuxième composant **tFlowMeter** et paramétrez ses propriétés.

- Cochez la case **Use input connection name as label**.
- Dans le champ **Mode**, sélectionnez l'option **Relative** et dans la liste déroulante **Reference connection**, sélectionnez *US_States* comme référence.
- Une fois de plus, aucun seuil n'est utilisé pour cet exemple, ainsi laissez le champ **Threshold** vide.
- Aucun paramètre particulier n'est requis pour le composant **tLogRow**.
- Il en va de même pour le composant **tFlowMeterCatcher** puisque ses propriétés sont limitées au schéma prédéfini contenant les informations de log.
- Paramétrez les propriétés du composant de sortie (**tFileOutputCSV**).

- Cochez la case **Append** pour logger toutes les mesures du **tFlowMeter**.
- Enregistrez le Job et exécutez-le.

```
Starting job FlowMeterCatcher at 17:56 29/08/2007.
19|Maine
20|Maryland
21|Massachusetts
22|Michigan
23|Minnesota
24|Mississippi
25|Missouri
26|Montana
Job FlowMeterCatcher ended at 17:56 29/08/2007. [exit code=0]
```

La vue **Run** affiche le nom des Etats filtrés tel qu'il a été défini dans le Job.

	A	K	L	M	N	O
1	moment	origin	label	count	reference	thresholds
2	08/29/07 05:56 PM	tFlowMeter_1	US_States	50		
3	08/29/07 05:56 PM	tFlowMeter_2	filtered_states	8	50	

Dans le fichier csv délimité, le nombre de lignes de la colonne **count** varie entre le composant **tFlowMeter1** et le **tFlowMeter2** puisque les données ont été filtrées. La colonne **reference** affiche cette différence.

tLogCatcher

Propriétés du tLogCatcher

Les composants **tDie** et **tWarn** sont étroitement liés au composant **tLogCatcher**. Ils sont généralement utilisés ensemble afin que les données de log collectées par le **tLogCatcher** soient rassemblées et envoyées vers la sortie définie.

Famille de composant	Logs & Errors	
Fonction	Le composant **tLogCatcher** récupère les champs et les messages des fonctions Java Exception et des composants **tDie** et/ou **tWarn** et les transmet au composant suivant.	
Objectif	Ce composant fonctionne comme une fonction de log déclenchée par une des trois fonctions suivantes : Java exception, **tDie** ou **tWarn**, pour collecter et transférer des données de log.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule.
	Catch Java Exception	Cochez cette case pour déclencher la fonction tCatch lorsque la fonction Java Exception est appelée dans le Job.
	Catch tDie	Cochez cette case pour déclencher la fonction tCatch lorsque le composant **tDie** est appelé dans le Job.
	Catch tWarn	Cochez cette case pour déclencher la fonction tCatch lorsque le composant **tWarn** est appelé dans le Job.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est le composant de début d'un Job secondaire qui se déclenche automatiquement à la fin du Job principal.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Avertissement & log lors de la création d'entrées

Dans ce scénario constitué de trois composants, un composant **tRowGenerator** crée des données aléatoires (des id à incrémenter). Le flux d'entrée atteint le composant **tWarn** qui déclenche le sous-Job commençant par le

tLogCatcher. Ce sous-Job récupère le message d'avertissement ainsi que les informations standard et les transmet au **tLogRow** afin d'afficher les données de log.

- Cliquez et déposez les composants **tRowGenerator**, **tWarn**, **tLogCatcher** et **tLogRow** dans l'espace de modélisation.
- Connectez les composants **tRowGenerator** et **tWarn**.
- Connectez les composants **tLogCatcher** et **tLogRow** séparément.
- Dans l'éditeur du **tRowGenerator**, paramétrez la création des données aléatoires à l'aide de fonction de base :

- Dans la vue **Component** du composant **tWarn**, paramétrez le message d'avertissement dans le champ **Warn message**, le code et le niveau de priorité dans le champ **Priority**. Pour cet exemple, le message est "this is a warning".
- Pour ce scénario, concaténez le message ci-dessus avec une fonction pour collecter la première valeur de la table Input.

- Dans la vue **Component** du composant **tLogCatcher**, cochez la case **tWarn** pour que le message défini dans le **tWarn** soit récupéré par le sous-Job.
- Cliquez sur **Edit Schema** pour vérifier le schéma utilisé pour la sortie de log. Notez que le log est complet.

Appuyez sur **F6** pour exécuter le Job. Notez que le Log produit est exhaustif.

Scénario 2 : Monitorer & arrêter un Job

Ce scénario utilise les composants **tLogCatcher** et **tDie**. Un composant **tRowGenerator** est connecté à un composant **tFileOutputDelimited** à l'aide d'un lien **Row**. Lorsqu'une erreur survient, le composant **tDie** déclenche le sous-Job contenant le **tLogCatcher** qui affiche les données de log dans la console **Run**.

- Cliquez et déposez les composants nécessaires de la **Palette** : **tRowGenerator**, **tFileOutputDelimited**, **tDie**, **tLogCatcher**, **tLogRow**.
- Dans la vue **Component** du composant **tRowGenerator**, paramétrez les propriétés des données d'entrée qui seront traitées.

- Editez le schéma et paramétrez les colonnes suivantes : *id*, *name*, *quantity*, *flag* et *creation*.
- Dans le champ **Number of rows**, saisissez 0. Cela constituera l'erreur sur laquelle sera basée opération Die.
- Dans la colonne **Values**, définissez les fonctions permettant d'alimenter le flux d'entrée.
- Paramétrez les propriétés du **tFileOutputDelimited** afin qu'il contienne les données transmises. La connexion **Row** provenant du **tRowGenerator** alimente automatiquement le schéma du **tFileOutputDelimited**. Dans le champ **Row separator**, saisissez un point-virgule comme séparateur de champ.

- Connectez le composant de sortie au **tDie** à l'aide d'un lien **Trigger** > **If**. Double-cliquez sur cette nouvelle connexion et paramétrez le if :

```
((Integer)globalMap.get("tRowGenerator_1_NB_LINE"))
<=0
```

- Puis double-cliquez pour sélectionner et paramétrer les propriétés du composant **tDie** dans l'onglet **Basic settings**.

- Dans le champ **Die message**, saisissez le message à transmettre au **tLogCatcher** avant que le Job ne soit arrêté par la fonction Kill.
- Dans le même Job, cliquez et déposez un **tLogCatcher**, sans pour autant le connecter aux composants précédents, et connectez-le à un composant **tLogRow**.
- Dans l'onglet **Basic settings** du composant **tLogCatcher**, paramétrez ses propriétés. Assurez-vous que la case **tDie** est bien cochée pour pouvoir ajouter un message Die aux informations de Log transmises au composant final.

```
Starting job tLogCatcher2 at 10:10 20/04/2010.

No row defined
2010-04-20
10:10:57|RPa7uU|RPa7uU|RPa7uU|TDQ_EE_MPX_JAVA|tLogCatcher2|Default|5|tDie|tDie_
1|No row defined|4
Job tLogCatcher2 ended at 10:10 20/04/2010. [exit code=4]
```

- Appuyez sur **F6** pour exécuter le Job et notez que le log contient un message en noir et un en rouge.
- Les données de log en noir proviennent du **tDie** et sont transmises par le **tLogCatcher**. Le message Java Exception en rouge apparaît puisque le Job s'est arrêté.

tLogRow

Propriétés du tLogRow

Famille de composant	Logs & Errors	
Fonction	Le composant **tLogRow** affiche les données ou les résultats dans la console **Run**.	
Objectif	Ce composant permet de monitorer les données traitées.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Cliquez pour synchroniser le fichier de sortie avec le schéma du fichier d'entrée. Cette fonction n'est disponible que quand le composant est relié au composant précédent via un lien **Row**.
	Basic	Affiche le flux de sortie en mode basique.
	Table	Affiche le flux de sortie dans des cellules de table.
	Vertical	Affiche chaque ligne du flux de sortie comme une liste de valeurs clé. Quand cette option est activée, vous pouvez, pour chaque ligne de sortie, choisir entre le nom unique, le nom du composant et les deux.
	Separator (mode basique uniquement)	Saisissez le séparateur qui délimitera les données dans la console.
	Print header (mode basique uniquement)	Activez cette option pour inclure l'en-tête du flux d'entrée dans l'affichage de sortie.
	Print component unique name in front of each output row (mode basique uniquement)	Cochez cette case si plusieurs composants **tLogRow** sont utilisés. Cela permet de différencier les sorties.

	Print schema column name in front of each value (mode basique uniquement)	Cochez cette case pour récupérer les libellés des colonnes du schéma de sortie.
	Use fixed length for values (mode basique uniquement)	Cochez cette case pour paramétrer une largeur fixe aux valeurs affichées.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en composant intermédiaire dans un flux de données ou en composant de fin dans un Job design.	
Utilisation dans les Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez également utiliser ce composant en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme étape intermédiaire ou finale. Il génère du code Map/Reduce natif qui peut être exécuté directement dans Hadoop. Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Utilisation dans des Jobs Storm	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez également utiliser ce composant en tant que composant Storm. Dans un Job Storm **Talend**, ce composant est utilisé comme étape intermédiaire ou finale. Il génère nativement du code Storm pouvant être exécuté directement dans un système Storm. La version de Storm ne supporte pas l'utilisation des variables globales. Vous devez utiliser l'onglet **Storm Configuration** dans la vue **Run** pour définir la connexion à un système Storm donné pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant un Job Storm **Talend**, consultez les sections décrivant comment créer et configurer un Job Storm **Talend** dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf lorsqu'explicitement indiqué, un scénario présente uniquement des Jobs de type **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour plus d'informations sur l'utilisation du **tLogRow**, consultez les composants :

- **tFileInputDelimited,** *Scénario : Lire les données maître d'un hub MDM*.
- **tContextLoad**, *Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement*.
- **tWarn**, **tDie** et **tLogCatcher**, *Scénario 1 : Avertissement & log lors de la création d'entrées* et *Scénario 2 : Monitorer & arrêter un Job*.

tStatCatcher

Propriétés du tStatCatcher

Famille de composant	Logs & Errors	
Fonction	Le composant **tStatCatcher** est basé sur le schéma prédéfini et regroupe les métadonnées du Job en cours d'exécution, ainsi que les métadonnées de chaque composant de ce Job lorsque la case **tStatCatcher Statistics** est cochée.	
Objectif	Ce composant regroupe les métadonnées du Job en cours d'exécution, ainsi que les métadonnées de chaque composant de ce Job lorsque la case **StatsCatcher Statistics** est cochée. Il collecte et transfère les données de log vers le composant suivant afin qu'elles soient affichées ou stockées.	
Basic settings	*Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Pour ce composant, le schéma est en lecture seule, puisqu'il regroupe les informations de log standard, notamment :
		Moment : Date et heure du traitement
		Pid : Identifiant système du Job.
		Father_pid : Identifiant système du Job parent, le cas échéant. Sinon, le Pid est dupliqué.
		Root_pid : Identifiant système du Job racine, le cas échéant. Sinon, le Pid est dupliqué.
		System_pid: Identifiant du process.
		Project : Nom du projet auquel appartient le Job.
		Job : Nom du Job en cours.
		Job_repository_id: Identifiant du fichier .item du Job stocké dans le référentiel.
		Job_version: Version du Job en cours.
		Context : Nom du contexte en cours.
		Origin : Nom du composant, le cas échéant.
		Message_type: Informations de début ou de fin du Job.
		Message : Informations de réussite ou d'échec du Job.
		Duration: Temps d'exécution d'un Job, ou d'un composant si la case **tStatCatcher Statistics** est cochée.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est le composant de début d'un Job secondaire qui se déclenche automatiquement à la fin du Job principal. Les dates de début et de fin apparaissent dans le log.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Afficher les statistiques d'exécution du Job

Ce scénario collecte les statistiques d'exécution du Job et les affiche dans la console **Run**. Notez que, comme aucune case **tStatCatcher Statistics** n'a été cochée pour les composants, les statistiques ne s'appliquent qu'à ce Job.

Relier les composants

1. Déposez un **tFixedFlowInput**, un **tFileOutputDelimited**, un**tStatCatcher** et un **tLogRow** dans l'espace de modélisation graphique.
2. Reliez le composant **tFixedFlowInput** au **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main**.
3. Reliez le composant **tStatCatcher** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **Edit schema** pour ouvrir l'éditeur de schéma.

3. Cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes, nommées respectivement *ID_Owners*, *Name_Customer* et *ID_Insurance*, de type **Integer** et **String**.

4. Cliquez sur **OK** pour valider la configuration et fermer l'éditeur.

5. Dans la fenêtre qui apparaît, cliquez sur **Yes** pour accepter la propagation des modifications.

6. Sélectionnez l'option **Use Inline Content (delimited file)**.

7. Dans le champ **Content**, saisissez `1;Andrew;888`.

8. Double-cliquez sur le **tFileOutputDelimited** pour ouvrir sa vue **Basic settings**.

9. Dans le champ **File Name**, saisissez le chemin d'accès complet du fichier vers lequel vous souhaitez sauvegarder les statistiques.

10. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

11. Sélectionnez l'option **Vertical (each row is a key/value list)** pour un affichage optimisé des résultats.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.

```
.-------------------------------------------.
|                #1. tLogRow_3              |
+-------------------+-----------------------+
| key               | value                 |
+-------------------+-----------------------+
| moment            | 2013-05-13 14:35:52   |
| pid               | pKWrWl                |
| father_pid        | pKWrWl                |
| root_pid          | pKWrWl                |
| system_pid        | 9828                  |
| project           | PP                    |
| job               | tStatCatcher          |
| job_repository_id | _TQ0rcK8cEeKOQccf-x19aQ |
| job_version       | 0.1                   |
| context           | Default               |
| origin            | null                  |
| message_type      | begin                 |
| message           | null                  |
| duration          | null                  |
+-------------------+-----------------------+
.-------------------------------------------.
|                #1. tLogRow_3              |
+-------------------+-----------------------+
| key               | value                 |
+-------------------+-----------------------+
| moment            | 2013-05-13 14:35:52   |
| pid               | pKWrWl                |
| father_pid        | pKWrWl                |
| root_pid          | pKWrWl                |
| system_pid        | 9828                  |
| project           | PP                    |
| job               | tStatCatcher          |
| job_repository_id | _TQ0rcK8cEeKOQccf-x19aQ |
| job_version       | 0.1                   |
| context           | Default               |
| origin            | null                  |
| message_type      | end                   |
| message           | success               |
| duration          | 66                    |
+-------------------+-----------------------+
```

Comme affiché ci-dessus, les statistiques d'exécution du Job ont été correctement générées.

tWarn

Propriétés du tWarn

Les composants **tDie** et **tWarn** sont étroitement liés au composant **tLogCatcher**. Ils sont généralement utilisés ensemble afin que les données de log collectées par le **tLogCatcher** soient rassemblées et envoyées vers la sortie définie.

Famille de composant	Logs & Errors	
Fonction	Le composant **tWarn** envoie un message d'avertissement au composant suivant. En cas d'erreur, il n'arrête pas l'exécution du Job. Si vous souhaitez arrêter le Job en cours en cas d'erreur, consultez *tDie*.	
Objectif	Ce composant déclenche un avertissement souvent pris comme données de log exhaustives par le composant **tLogCatcher**.	
Basic settings	*Warn message*	Saisissez votre message d'avertissement.
	Code	Saisissez le niveau de code.
	Priority	Sélectionnez le niveau de priorité.
Global Variables	**WARN_MESSAGES** : message d'avertissement. Cette variable est une variable **After** et retourne une chaîne de caractères. **WARN_CODE** : niveau de code du message d'avertissement. Cette variable est une variable **After** et retourne un nombre entier. **WARN_PRIORITY** : niveau de priorité du message d'avertissement. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant ne peut être utilisé comme composant de début. Ainsi, s'il est connecté à un composant de sortie, il doit aussi être connecté à un composant d'entrée.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénarios associés

Pour des exemples d'utilisation du **tWarn**, consultez les scénarios du composant **tLogCatcher** :

- *Scénario 1 : Avertissement & log lors de la création d'entrées*.
- *Scénario 2 : Monitorer & arrêter un Job*.

Composants Misc

Ce chapitre passe en revue les principaux composants présents dans la famille **Misc** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **Misc** regroupe divers connecteurs permettant de répondre à des besoins, tels que la création de lignes de données factices, la mise en tampon de données, le chargement de variables de contexte.

tAddLocationFromIP

Propriétés du tAddLocationFromIP

Famille du composant	Misc	
Fonction	Le composant **tAddLocationFromIP** remplace des adresses IP par le lieu géographique correspondant.	
Objectif	Le **tAddLocationFromIP** vous permet de localiser géographiquement les visiteurs grâce à leurs adresses IP : il identifie le lieu géographique des visiteurs (pays, région, ville, latitude, longitude, code postal, etc.) en utilisant un fichier de consultation d'adresse IP.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit les champs à traiter et à passer au
		Built-in : Le schéma est créé et conservé pour ce composant uniquement. Voir également le Guide utilisateur du Studio Talend.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Database Filepath	Chemin d'accès au fichier de consultation d'adresses IP.
	Input parameters	**Input column :** Sélectionnez la colonne d'entrée à partir de laquelle les valeurs sont collectées pour l'agrégation.
		input value is a hostname : Cochez cette case si la colonne d'entrée comporte des noms d'hôtes.
		input value is an IP address : Cochez cette case si la colonne d'entrée comporte des adresses IP.
	Location type	**Country code** : Cochez cette case pour remplacer l'adresse IP par le code du pays correspondant.
		Country name : Cochez cette case pour remplacer l'adresse IP par le nom du pays correspondant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé comme étape intermédiaire du traitement permettant de remplacer des adresses IP par des informations géolocalisées. Il ne peut être utilisé comme composant de début car il nécessite un flux en entrée, ainsi qu'un composant de sortie.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

	• geoip.jar

Scénario : Identifier la localisation géographique d'une adresse IP

Le scénario Java suivant crée un Job à trois composants associant une adresse IP à un lieu géographique. Il permet d'obtenir la localisation géographique du visiteur d'un site à partir de son adresse IP.

Déposer et relier les composants

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l'éditeur graphique : le **tFixedFlowInput**, le **tAddLocationFromIP** et le **tLogRow**.

2. Connectez les trois composants en utilisant des liens de type **Row Main**.

Configurer les composants

1. Dans l'espace graphique, sélectionnez le **tFixedFlowInput** et cliquez sur l'onglet **Component** pour définir la configuration de base (**Basic settings**) du **tFixedFlowInput**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit Schema** pour définir les données que vous voulez utiliser en entrée. Dans ce scénario, le schéma est constitué d'une colonne comportant une adresse IP.

3. Cliquez sur le bouton **OK** pour fermer la boîte de dialogue, puis acceptez la propagation des modifications lorsque le système vous invite à le faire. La colonne définie s'affiche dans la section **Values** de la vue **Basic settings**.

4. Dans le champ **Number of rows**, saisissez le nombre de lignes à générer puis cliquez dans la cellule **Value** et définissez la valeur de l'adresse IP.

5. Dans l'espace de modélisation graphique, sélectionnez le composant **tAddLocationFromIP** et cliquez dans la vue **Component** pour définir la configuration de base (**Basic settings**) du **tAddLocationFromIP**.

6. Cliquez sur le bouton **Sync columns** pour synchroniser le schéma avec le schéma d'entrée du **tFixedFlowInput**.

7. Parcourez vos dossiers jusqu'au fichier *GeoIP.dat* pour en définir le chemin d'accès dans le champ **Database filepath**.

   Assurez-vous de télécharger la dernière version du fichier de consultation des adresses IP, à partir du site indiqué dans la vue **Basic settings** du composant **tAddLocationFromIp**.

8. Dans la zone **Input parameters**, définissez les paramètres d'entrée selon vos besoins. Dans ce scénario, la colonne d'entrée correspond à la colonne *ip*, qui comporte une adresse IP, définie précédemment.

9. Dans la zone **Location type**, définissez le type de lieu selon vos besoins. Dans ce scénario, vous allez afficher le nom du pays.

10. Dans l'éditeur graphique, sélectionnez le composant **tLogRow**. Cliquez sur la vue **Component** puis définissez la configuration de base (**Basic settings**) du **tLogRow** selon vos besoins. Dans ce scénario, vous allez afficher les valeurs dans les cellules d'une table.

11. Sauvegardez votre Job puis cliquez sur **F6** pour l'exécuter.

```
Starting job add_location at 11:00 06/08/2008.
.------------+--------.
|      tLogRow_1       |
|=-----------+-------=|
|ip          |location|
|=-----------+-------=|
|88.131.8.250|Sweden  |
'------------+--------'

Job add_location ended at 11:00 06/08/2008. [exit code=0]
```

La ligne unique générée par le Job affiche le nom du pays correspondant à l'adresse IP renseignée.

tBufferInput

Propriétés du tBufferInput

Famille de composant	Misc	
Fonction	Ce composant permet de récupérer les données mises en tampon pour pouvoir les réutiliser dans un deuxième sous-job.	
Objectif	Le composant **tBufferInput** récupère les données mises en mémoire tampon, via un composant **tBufferOutput** par exemple, pour les traiter dans un deuxième temps.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Lors de l'utilisation du **tBufferInput**, c'est l'ordre des colonnes qui est pris en compte, ainsi l'ordre des colonnes est plus important que leur libellé.
		Built-in : Le schéma est créé et conservé pour ce composant uniquement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est le composant de début d'un Job secondaire qui se déclenche automatiquement à la fin du Job principal.	

Scénario : Récupérer les données mises en mémoire tampon

Ce scénario décrit un Job qui récupère les données mises en mémoire tampon lors de l'exécution d'un premier sous-job et les affiche dans la console standard.

- Cliquez et déposez les composants suivants : **tFileInputDelimited** et **tBufferOutput**.
- Sélectionnez le composant **tFileInputDelimited** et dans sa vue **Component**, paramétrez les propriétés d'accès au fichier d'entrée.

- Dans le champ **File Name**, paramétrez le chemin d'accès au fichier délimité contenant les données à mettre en tampon.
- Dans les champs **Row Separator** et **Field Separator**, renseignez respectivement les séparateurs de lignes et de champs.
- Dans le champ **Header**, renseignez les informations sur un éventuel en-tête.
- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour décrire la structure du fichier

- Décrivez le **Schéma** des données à passer au composant **tBufferOutput**.

- Sélectionnez le composant **tBufferOutput** et paramétrez ses propriétés dans la vue **Component**.

- Généralement le schéma du composant d'entrée alimente automatiquement le schéma du composant **tBufferOutput**. Mais vous pouvez aussi configurer une partie du schéma à mettre en tampon si vous le souhaitez.
- Déposez les composants **tBufferInput** et **tLogRow** en dessous du sous-job que vous venez de créer.
- Reliez le composant **tFileInputDelimited** au **tBufferInput** via un lien de type **Trigger** > **OnSubjobOk** et reliez le composant **tBufferInput** au **tLogRow** via un lien de type **Row** > **Main**.
- Double-cliquez sur le composant **tBufferInput** pour paramétrer ses propriétés dans la vue **Component**.
- Dans l'onglet **Basic settings**, cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour décrire la structure du fichier.

- Utilisez la même description de fichier que celle définie pour le composant **tFileInputDelimited** et cliquez sur **OK**.
- Le schéma du composant **tBufferInput** alimente automatiquement le schéma du composant **tLogRow**. Sinon double-cliquez sur le **tLogRow** pour afficher ses propriétés dans la vue **Component** et cliquez sur le bouton **Sync columns**.
- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job tBufferInput_scenario at 14:13 12/06/2009.

AL|Alabama|Montgomery|Birmingham
AK|Alaska|Juneau|Anchorage
AZ|Arizona|Phoenix|Phoenix
AR|Arkansas|Little Rock|Little Rock
CA|California|Sacramento|Los Angeles
CO|Colorado|Denver|Denver
CT|Connecticut|Hartford|Bridgeport
DE|Delaware|Dover|Wilmington
FL|Florida|Tallahassee|Jacksonville
GA|Georgia|Atlanta|Atlanta
HI|Hawaii|Honolulu|Honolulu
ID|Idaho|Boise|Boise
IL|Illinois|Springfield|Chicago
IN|Indiana|Indianapolis|Indianapolis
IA|Iowa|Des Moines|Des Moines
KS|Kansas|Topeka|Wichita
KY|Kentucky|Frankfort|Louisville
```

La console affiche les données qui ont été récupérés de la mémoire tampon.

tBufferOutput

Propriétés du tBufferOutput

Famille de composant	Misc	
Fonction	Ce composant met en tampon des données afin de pouvoir y accéder plus tard via un service Web par exemple.	
Objectif	Ce composant permet d'accéder aux données via un service Web. En effet, il a été conçu pour être exporté en tant que service Web afin d'accéder aux données directement à partir du serveur d'application Web. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Lors de l'utilisation du **tBufferOutput**, c'est l'ordre des colonnes qui est pris en compte, ainsi l'ordre des colonnes est plus important que leur libellé.
		Built-in : Le schéma est créé et conservé pour ce composant uniquement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant n'est pas un composant de début (font vert) et il requiert un composant de sortie.	

Scénario 1 : Mettre des données en mémoire tampon

Ce scénario décrit un Job volontairement basique qui met en tampon les données d'un Job enfant pendant qu'un Job parent affiche les données mise en tampon dans la console standard. Généralement, un Job utilisant un composant **tBufferOutput** inclurait une étape d'export en tant que service Web, afin que les données soient accessibles directement via le serveur d'application Web.

- Créez deux Jobs : le premier Job (*BufferFatherJob*) exécute le deuxième et affiche le contenu dans la console **Run**. Le deuxième Job (*BufferChildJob*) stocke les données définies dans une mémoire tampon.
- Dans le premier Job, cliquez et déposez les composants suivants : **tRunJob** et **tLogRow**.
- Dans le deuxième Job, cliquez et déposez les composants suivants : **tFileInputDelimited** et **tBufferOutput**.
- Commencez par paramétrer les propriétés du deuxième Job.
- Sélectionnez le composant **tFileInputDelimited** et dans l'onglet **Basic settings** de la vue **Component**, paramétrez les propriétés d'accès au fichier d'entrée.

- Dans le champ **File Name**, paramétrez le chemin d'accès au fichier délimité contenant les données à mettre en tampon.
- Dans les champs **Row Separator** et **Field Separator**, renseignez respectivement les séparateurs de lignes et de champs.
- Dans le champ **Header**, renseignez les informations sur un éventuel en-tête.

tFileInputDelimited_2

Column	Key	Type	Nullable	Date Patt...	Length	Pre...	D...	Co...
Postal	☐	String	☑		2			
State	☐	String	☑		14			
Capital	☐	String	☑		14			
MostPopulousCity	☐	String	☑		14			

- Décrivez le **Schema** des données à passer au composant **tBufferOutput**.
- Sélectionnez le composant **tBufferOutput** et paramétrez ses propriétés dans l'onglet **Basic settings** de la vue **Component**.

Schema Type [Built-In] Edit schema [...] [Sync columns]

- Généralement le schéma du composant d'entrée alimente automatiquement le schéma du composant **tBufferOutput**. Mais vous pouvez aussi configurer une partie du schéma à mettre en tampon si vous le souhaitez.
- Maintenant le premier Job (*BufferFatherJob*), paramétrez les propriétés du composant **tRunJob**.

Schema Type [Built-In] Edit schema [...]

☑ Die on child error

Job [/v23a/Buffering/BufferChildJob] * Context [Defa

Context Param

Parameters	Values

- Cliquez sur **Edit schema** pour modifier le schéma si nécessaire et sélectionner la colonne à afficher. Le schéma peut être identique ou différent du schéma mis en tampon.
- Dans le tableau **Context Param**, vous pouvez aussi définir les paramètres de contexte à utiliser pour l'exécution de ce Job. Pour cet exemple, gardez le contexte par défaut.

Appuyez sur **F6** pour exécuter les Jobs parent. Le composant **tRunJob** se charge d'exécuter le Job enfant et affiche les données dans la console standard :

```
Starting job BufferFatherJob at 10:41 05/02/2008.
AL|Alabama|Montgomery|Birmingham
AK|Alaska|Juneau|Anchorage
AZ|Arizona|Phoenix|Phoenix
AR|Arkansas|Little Rock|Little Rock
CA|California|Sacramento|Los Angeles
CO|Colorado|Denver|Denver
CT|Connecticut|Hartford|Bridgeport
DE|Delaware|Dover|Wilmington
FL|Florida|Tallahassee|Jacksonville
GA|Georgia|Atlanta|Atlanta
HI|Hawaii|Honolulu|Honolulu
ID|Idaho|Boise|Boise
IL|Illinois|Springfield|Chicago
IN|Indiana|Indianapolis|Indianapolis
IA|Iowa|Des Moines|Des Moines
KS|Kansas|Topeka|Wichita
KY|Kentucky|Frankfort|Louisville
```

Scénario 2 : Mettre les données de sortie en mémoire tampon du serveur d'application Web

Ce scénario décrit un Job appelant un service Web et stockant les données de sortie directement sur la mémoire tampon du serveur d'application Web. Ce scénario crée d'abord un Job utilisant des variables de contexte qui pourra ensuite être utilisé en tant que service Web, puis exporte de Job en tant que service Web.

Créer un Job

1. Cliquez-déposez les composants suivants dans le Job Designer : **tFixedFlowInput** et **tBufferOutput**.
2. Reliez-les via une connexion de type **Row Main**.

Créer une variable de contexte

Dans ce scénario, vous allez définir deux variables de contexte : *nb_lines* et *lastname*. La première variable configure le nombre de lignes générées par le composant **tFixedFlowInput** et la seconde configure le nom de famille à afficher dans la liste de sortie. Pour plus d'informations concernant la création et l'utilisation de variables de contexte, consultez le *Guide utilisateur de Studio Talend*.

Pour définir les deux variables de contexte :

1. Dans l'onglet **Contexts** de votre Job, cliquez sur le bouton **[+]** au bas de la vue, afin d'ajouter deux variables, respectivement *nb_lines*, de type **Integer** et *lastname* de type **String**.

2. Dans le champ **Value** des variables, configurez le nom de famille à afficher et le nombre de lignes à générer, respectivement *Ford* et *3* dans cet exemple.

Configurer les données d'entrée

1. Dans l'espace de travail, sélectionnez le **tFixedFlowInput**.

2. Cliquez sur la vue **Component** pour paramétrer ses propriétés de base dans l'onglet **Basic settings**.

3. Cliquez sur le bouton **[...]** à côté du champ **Edit Schema** pour décrire la structure des données que vous souhaitez à partir des variables internes. Dans ce scénario, le schéma est constitué de trois colonnes : *now*, de type **Date**, *firstname* et *lastname*, toutes deux de type **String**.

4. Cliquez sur **OK** pour fermer la boîte de dialogue et acceptez la propagation des modifications lorsque vous y êtes invité. Les trois colonnes définies apparaissent dans le champ **Values** de l'onglet **Basic settings** du **tFixedFlowInput**.

5. Cliquez dans la cellule **Value** de chaque ligne des deux premier paramètres définis et appuyez sur **Ctrl +Espace** pour accéder à la liste des variables globales.

6. A partir de la liste des variables globales, sélectionnez *TalendDate.getCurrentDate()* pour la colonne *now* et *talendDatagenerator.getFirstName* pour la colonne *firstname*.

7. Cliquez dans la cellule **Value** de la colonne *lastname* et appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales.

8. A partir de la liste des variables globales, sélectionnez *context.lastname*, la variable de contexte que vous avez créé pour la colonne *lastname*.

Construire un Job en tant que service Web

Avant de construire votre Job en tant que service Web, consultez le *Guide utilisateur du Studio Talend*.

1. Dans la vue **Repository**, cliquez-droit sur le Job que vous venez de créer et sélectionnez **Build Job** dans le menu. La boîte de dialogue **[Build Job]** s'ouvre.

2. Cliquez sur le bouton **Browse...** et sélectionnez le répertoire dans lequel enregistrer votre Job.

3. Dans la zone **Build type**, sélectionnez le type de construction (build) que vous souhaitez utiliser dans votre application Web Tomcat (WAR pour cet exemple) et cliquez sur **Finish**. La boîte de dialogue **[Build Job]** se referme.

4. Copiez le fichier War et collez-le dans le répertoire de votre Tomcat.

Scénario 3 : Appeler un Job contenant des variables de contexte à partir de votre navigateur Web

Ce scénario décrit la procédure à utiliser pour appeler le Job créé dans *Scénario 2 : Mettre les données de sortie en mémoire tampon du serveur d'application Web* à partir de votre navigateur Web en modifiant/sans modifier les valeurs des variables de contexte.

Saisissez l'URL suivante dans votre navigateur Web : *http://localhost:8080//export_job/services/export_job3?method=runJob* dans laquelle "export_job" correspond au nom du répertoire de votre application Web déployé sur le Tomcat et "export_job3" correspond au nom de votre Job.

Cliquez sur **Enter** pour exécuter votre Job à partir du navigateur.


```
- <soapenv:Envelope>
  - <soapenv:Body>
    - <runJobReturn xsi:type="ns1:runJobReturn">
      - <ns1:item xsi:type="ns1:ArrayOf_xsd_string">
          <ns1:item xsi:type="xsd:string">31-07-2008</ns1:item>
          <ns1:item xsi:type="xsd:string">William</ns1:item>
          <ns1:item xsi:type="xsd:string">Ford</ns1:item>
        </ns1:item>
      - <ns1:item xsi:type="ns1:ArrayOf_xsd_string">
          <ns1:item xsi:type="xsd:string">31-07-2008</ns1:item>
          <ns1:item xsi:type="xsd:string">Millard</ns1:item>
          <ns1:item xsi:type="xsd:string">Ford</ns1:item>
        </ns1:item>
      - <ns1:item xsi:type="ns1:ArrayOf_xsd_string">
          <ns1:item xsi:type="xsd:string">31-07-2008</ns1:item>
          <ns1:item xsi:type="xsd:string">James</ns1:item>
          <ns1:item xsi:type="xsd:string">Ford</ns1:item>
        </ns1:item>
      </runJobReturn>
    </soapenv:Body>
  </soapenv:Envelope>
```


Le Job utilise les valeurs par défaut des variables de contexte : *nb_lines* et *lastname*, et donc génère trois lignes contenant la date en cours, un prénom et le nom Ford.

Vous pouvez modifier les valeurs des variables de contexte directement à partir de votre navigateur. Pour appeler le Job à partir de votre navigateur et modifier les valeurs des deux variables de contexte, saisissez l'URL suivante :

http://localhost:8080//export_job/services/export_job3?method=runJob&arg1=--context_param %20lastname=MASSY&arg2=--context_param%20nb_lines=2.

%20 correspond à un espace blanc lorsqu'il est utilisé dans une URL. Dans le premier argument "arg1", définissez la valeur "MASSY" pour la variable de contexte *lastname*. Ainsi le nom qui apparaîtra sera "MASSY". Dans le deuxième argument "arg2", définissez la valeur "2" pour la variable de contexte *nb_lines* pour ne générer que deux lignes de données.

Cliquez sur **Enter** pour exécuter le Job à partir du navigateur.

```
- <soapenv:Envelope>
  - <soapenv:Body>
    - <runJobReturn xsi:type="ns1:runJobReturn">
      - <ns1:item xsi:type="ns1:ArrayOf_xsd_string">
          <ns1:item xsi:type="xsd:string">31-07-2008</ns1:item>
          <ns1:item xsi:type="xsd:string">Richard</ns1:item>
          <ns1:item xsi:type="xsd:string">MASSY</ns1:item>
        </ns1:item>
      - <ns1:item xsi:type="ns1:ArrayOf_xsd_string">
          <ns1:item xsi:type="xsd:string">31-07-2008</ns1:item>
          <ns1:item xsi:type="xsd:string">Theodore</ns1:item>
          <ns1:item xsi:type="xsd:string">MASSY</ns1:item>
        </ns1:item>
      </runJobReturn>
    </soapenv:Body>
  </soapenv:Envelope>
```


Le Job génère deux lignes avec MASSY comme nom.

Scénario 4 : Appeler un Job exporté en tant que service Web dans un autre Job

Ce scénario décrit un Job appelant un autre Job exporté en tant que service Web en utilisant un composant **tWebServiceInput**. Ce scénario appellera le Job créé dans *Scénario 2 : Mettre les données de sortie en mémoire tampon du serveur d'application Web*.

- Cliquez-déposez les composants suivants dans le Job Designer : **tWebServiceInput** et **tLogRow**.
- Reliez-les via une connexion de type **Row Main**.

- Dans l'espace de travail, sélectionnez le composant **tWebServiceInput**.
- Cliquez sur la vue **Component** pour définir ses propriétés de base dans l'onglet **Basic settings**.

- Dans la liste **Schema**, sélectionnez **Built-In** et cliquez sur le bouton **[...]** à côté du champ **Edit Schema** pour décrire la structure des données que vous souhaitez appeler du Job exporté. Dans ce scénario, le schéma est constitué de trois colonnes, *now*, *firstname*, et *lastname*.

- Cliquez sur le bouton **[+]** pour ajouter les trois lignes de paramètres et définir vos variables. Cliquez sur **OK** pour fermer la boîte de dialogue.
- Dans le champ **WSDL** de l'onglet **Basic settings** de la vue **Component** du **tWebServiceInput**, saisissez l'URL : *http://localhost:8080/export_job/services/export_job3?WSDL* dans laquelle "export_job" correspond au nom du répertoire de votre application Web dans lequel le Job à appeler est stocké et "export_job3" correspond au nom du Job.

- Dans le champ **Method name**, saisissez *runJob*.
- Dans la zone **Parameters**, cliquez sur le bouton **[+]** pour ajouter deux lignes de paramètres et définir vos variables de contexte.
- Cliquez dans la première cellule **Value** pour configurer le paramètre définissant le nombre de lignes à générer : *--context_param nb_line=3*.
- Cliquez dans la deuxième cellule **Value** pour configurer le paramètre définissant le nom à afficher : *--context_param lastname=Ford*.
- Sélectionnez le **tLogRow** et cliquez sur la vue **Component** pour afficher les propriétés du composant.
- Dans la vue **Basic settings** du **tLogRow**, sélectionnez l'option **Table** pour afficher les données de sortie sous forme de tableau. Pour plus d'informations, consultez *Propriétés du tLogRow*.

- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job Call_Webservice_In_Job at 14:12 01/08/2008.
.----------+---------+--------.
|          tLogRow_1           |
|=---------+---------+-------=|
|now       |firstname|lastname|
|=---------+---------+-------=|
|01-08-2008|Ronald   |Ford    |
|01-08-2008|Woodrow  |Ford    |
|01-08-2008|Franklin |Ford    |
'----------+---------+--------'

Job Call_Webservice_In_Job ended at 14:12 01/08/2008. [exit code=0]
```

Le Job génère trois colonnes contenant la date en cours, des prénoms et des noms, et les affiche dans la console sous forme de tableau.

tContextDump

Propriétés du tContextDump

Famille de composant	Misc	
Fonction	**tContextDump** passe la configuration du contexte du Job courant au composant suivant.	
Objectif	Le **tContextDump** copie la configuration du contexte du Job courant dans un fichier plat, une table de base de données, etc, pouvant être réutilisé(e) par le **tContextLoad**. Utilisé avec le **tContextLoad**, ce composant permet d'appliquer simplement la configuration de contexte d'un Job à un autre.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit les champs qui sont traités et passés au composant suivant. Le schéma du **tContextDump** est en lecture seule et se compose de deux colonnes, *Key* et *Value*, qui correspondent au nom et à la valeur du paramètre du contexte du Job.
	Hide Password	Cochez cette case pour masquer la valeur du mot de passe du paramètre de contexte, c'est-à-dire pour afficher des * pour la valeur des paramètres de contexte dont le *Type* est *Password*.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant d'entrée et passe la configuration du contexte du Job à un fichier, une table d'une base de données, etc.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tContextLoad

Propriétés du tContextLoad

Famille de composant	Misc	
Fonction	Le composant **tContextLoad** modifie dynamiquement les valeurs du contexte actif.	
Objectif	Le **tContextLoad** permet de charger un contexte à partir d'un flux entrant. Ce composant effectue aussi deux contrôles. Il prévient lorsque les paramètres définis dans le flux entrant ne sont pas définis dans le contexte. Il avertit également lorsque la valeur d'un contexte n'est pas initialisée dans le flux entrant. Notez que ces avertissements ne bloquent pas le traitement.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champ qui sont traités et passés au composant suivant. Le schéma du composant **tContextLoad** doit être composé de deux colonnes, l'une contenant le nom du paramètre et la deuxième contenant la valeur du paramètre à charger.
	If a variable loaded, but not in the context	Si une variable est chargée mais n'apparaît pas dans le contexte, choisissez comment doit s'afficher la notification. Sous forme d'erreur (**Error**), de warning (**warning**), ou d'information (**info**).
	If a variable in the context, but not loaded	Si une variable apparaît dans le contexte mais n'est pas chargée, choisissez comment doit s'afficher la notification. Sous forme d'erreur (**Error**), de warning (**warning**), ou d'information (**info**).
	Print operations	Cochez cette case pour afficher les paramètres de contexte dans la vue **Run**.
	Disable errors	Cochez cette case pour ne pas afficher d'erreur.
	Disable warnings	Cochez cette case pour ne pas afficher de warning.
	Disable infos	Cochez cette case pour ne pas afficher d'information.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et les lignes contenant les erreurs seront ignorées.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin d'activer ou désactiver dynamiquement l'option **Print operations** au moment de l'exécution. Lorsqu'un paramètre dynamique est configuré, l'option **Print operations** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **KEY_NOT_INCONTEXT** : les variables sont chargées mais n'apparaissent pas dans le contexte. Cette variable est une variable **After** et retourne une chaîne de caractères. **KEY_NOT_LOADED** : les variables ne sont pas chargées mais apparaissent dans le contexte. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant s'appuie sur le flux de données pour charger les valeurs de contexte à utiliser. Ainsi, il requiert un composant d'entrée et ne peut pas être un composant de début.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Le **tContextLoad** ne crée aucune variable qui n'a pas été définie dans le contexte par défaut.

Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement

Le Job décrit dans ce scénario est composé de deux sous-jobs. Le premier permet de charger dynamiquement des paramètres de contexte à partir de deux fichiers texte. Le second utilise ces paramètres chargés afin de se connecter à deux bases de données différentes et d'afficher le contenu d'une table de base de données existante pour chacune de ces bases. Les paramètres de contexte du Job permettent de décider à quelle base de données se connecter et de choisir d'afficher ou non les paramètres de contexte définis dans la console à l'exécution.

Déposer et relier les composants

1. Déposez, depuis la **Palette**, un **tFileInputDelimeted** et un **tContextLoad** dans l'espace de modélisation graphique. Reliez-les à l'aide d'un lien **Row** > **Main** afin de créer le premier sous-job.
2. Déposez un **tMysqlInput** et un **tLogRow** dans l'espace de modélisation graphique. Reliez-les à l'aide d'un lien **Row** > **Main** afin de créer le second sous-job.
3. Reliez les deux sous-jobs à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Préparer les contextes et les variables de contexte

1. Créez deux fichiers délimités correspondant aux deux contextes de ce scénario, à savoir les accès aux deux bases de données et nommez-les respectivement *test_connection.txt* et *prod_connection.txt*. Ces fichiers contiennent les informations de connexion aux bases de données, respectivement pour le test et pour la production. Chaque fichier est composé de deux colonnes, contenant le nom des paramètres et leurs valeurs correspondantes. Voici un exemple :

```
host;localhost
port;3306
database;test
username;root
password;talend
```

2. Sélectionnez la vue **Contexts** puis cliquez sept fois sur le bouton **[+]** pour ajouter sept lignes à la table afin de définir les paramètres suivants :

   - *host*, de type **String**
   - *port*, de type **String**
   - *database*, de type **String**
   - *username*, de type **String**
   - *password*, type **Password**
   - *filename*, de type **File**
   - *printOperations*, de type **Boolean**.

   Notez que les paramètres *host*, *port*, *database*, *username* et *password* correspondent au nom des paramètres dans les fichiers délimités et sont utilisés pour configurer la connexion à la base de données souhaitée. Le paramètre *filename* est utilisé pour définir le fichier délimité à lire au moment de l'exécution du Job et le paramètre *printOperations* est utilisé pour décider d'afficher ou non les paramètres de contexte définis par le **tContextLoad** dans la console.

3. Cliquez sur l'onglet **Contexts** et cliquez sur le bouton **[+]** situé en haut à droite du panneau afin d'ouvrir la boîte de dialogue **[Configure Contexts]**.

4. Sélectionnez le contexte par défaut, cliquez sur le bouton **Edit** et renommez le contexte en *Test*.

5. Cliquez sur **New** afin d'ajouter un nouveau contexte nommé *Production*. Cliquez ensuite sur **OK** pour fermer la boîte de dialogue.

6. De retour dans l'onglet **Contexts**, définissez la valeur de la variable *filename* sous chaque contexte, en cliquant dans le champ **Value** respectif et en parcourant votre système jusqu'au fichier délimité.

7. Cochez la case **Prompt** près du champ **Value** de la variable *filename*, pour que les deux contextes affichent les champs **Prompt** et saisissez le message à afficher lors de l'exécution.

8. Pour la variable *printOperations*, cliquez dans le champ **Value** sous le contexte *Production* et sélectionnez **false** dans la liste. Cliquez dans le champ **Value** sous le contexte *Test* et sélectionnez **true** dans la liste. Cochez la case **Prompt** sous chaque contexte et saisissez le message à afficher lors de l'exécution.

	Name	Type	Production			Test		
			Value		Prompt	Value		Prompt
1	host	String	null	☐	host?	null	☐	host?
2	port	String	null	☐	port?	null	☐	port?
3	database	String	null	☐	database?	null	☐	database?
4	username	String	null	☐	username?	null	☐	username?
5	password	Password		☐	password?		☐	password?
6	filename	File	E:/Talend/Data/Input/pr...	☑	Specify the file:	E:/Talend/Data/Input/te...	☑	Specify the file:
7	printOperation	boolean | Bo...	false	☑	Show loaded variables?	true	☑	Show loaded variables

Default context environment Test

Configurer les composants

1. Dans le champ **File name/Stream** de la vue **Basic settings** du **tFileInputDelimited**, saisissez le nom de la variable de contexte correspondante définie précédemment : `context.filename`.

2. Définissez manuellement (**Built-in**) le schéma du fichier. Il contient deux colonnes : *key* et *value*.

3. Acceptez la propagation du schéma défini au composant suivant (**tContextLoad**).

4. Dans la vue **Dynamic settings** du **tContextLoad**, cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez `context.printOperations` afin d'utiliser la variable de contexte *printOperations* définie précédemment. Notez que la case **Print Operations** de la vue **Basic settings** devient grisée et inaccessible.

5. Double-cliquez sur le **tMysqlInput** afin d'ouvrir sa vue **Basic settings**.

6. Dans les champs **Host**, **Port**, **Database**, **Username** et **Password**, saisissez les variables stockées dans les fichiers délimités et définies dans la vue **Contexts** : `context.host`, `context.port`, `context.database`, `context.username`, et `context.password` respectivement dans cet exemple. Dans le champ **Table name**, saisissez le nom de la table de base de données à partir de laquelle lire les données, *customers* pour les deux bases de données de cet exemple.

7. Dans le champ **Schema**, saisissez les informations sur le schéma. Si vous avez stocké le schéma dans le nœud **Metadata** du **Repository**, sélectionnez-le après avoir sélectionné l'option **Repository**.

   Dans cet exemple, le schéma des deux bases de données est composé de quatre colonnes : *id* (de type **INT** et d'une longueur de 2 caractères), *firstname* (de type **VARCHAR** et d'une longueur de 15 caractères), *lastName* (de type **VARCHAR** et d'une longueur de 15 caractères) et *city* (de type **VARCHAR** et d'une longueur de 15 caractères).

8. Dans le champ **Query**, saisissez la requête SQL à exécuter sur la table de base de données spécifiée. Dans cet exemple, cliquez sur **Guess Query** afin de récupérer toutes les colonnes de la table. Elles s'affichent dans la vue **Run** grâce au **tLogRow**.

9. Dans la vue **Basic settings** du **tLogRow**, sélectionnez l'option **Table** afin d'afficher les enregistrements sous forme de tableau.

Exécuter le Job

1. Appuyez sur **Ctrl+S** afin de sauvegarder le Job. Appuyez sur **F6** afin d'exécuter le Job avec le contexte par défaut, *Test* dans ce scénario.

   Une boîte de dialogue s'affiche, vous demandant de spécifier le fichier délimité à lire et de décider d'afficher ou non les paramètres de contexte définis dans la console.

Si vous le souhaitez, vous pouvez spécifier un fichier autre que celui par défaut. Vous pouvez également décocher la case **Show loaded variables** si vous ne souhaitez pas voir les variables de contexte dans la console. Afin d'exécuter le Job avec les paramètres par défaut, cliquez sur **OK**.

Les paramètres de contexte et le contenu de la table de base de données correspondant au contexte *Test* s'affichent dans la console **Run**.

2. Sélectionnez à présent le contexte *Production* et appuyez sur **F6** afin d'exécuter le Job à nouveau. Lorsque la boîte de dialogue s'affiche, cliquez sur **OK** pour exécuter le Job avec les paramètres par défaut.

Le contenu de la table de base de données correspondant au contexte *Test* s'affiche dans la console **Run**. Comme la variable *printOperations* a la valeur *false*, les paramètres de contexte définis ne s'affichent pas dans la console.

tFixedFlowInput

Propriétés du tFixedFlowInput

Famille du composant	Misc	
Fonction	Le composant **tFixedFlowInput** génère autant de lignes et de champs que vous souhaitez alimenter par des valeurs fixes.	
Objectif	Le **tFixedFlowInput** permet de générer des données fixes à partir de variables internes.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Mode	Sélectionnez le mode, parmi les trois proposés, que vous souhaitez utiliser. **Use Single Table :** Saisissez la donnée que vous souhaitez générer dans la colonne Value correspondante. **Use Inline Table :** Ajoutez la ou les ligne(s) que vous souhaitez générer. **Use Inline Content :** Saisissez vos données à générer, séparées par les séparateurs préalablement définis dans les champs **Row** et **Field Separator**.
	Number of rows	Saisissez le nombre de lignes de données à générer.
	Values	Saisissez entre guillemets les valeurs correspondant aux colonnes définies dans la boîte de dialogue du schéma, via le bouton **Edit schema**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.	

	ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé comme composant de début ou de milieu, il requiert donc un composant de sortie.
Utilisation dans des Jobs Storm	Dans un Job Storm **Talend**, ce composant est utilisé comme composant de début. Les autres composants utilisés avec celui-ci doivent également être des composants Storm. Ils génèrent nativement du code Storm pouvant être exécuté directement dans un système Storm. La version de Storm ne supporte pas l'utilisation des variables globales. Vous devez utiliser l'onglet **Storm Configuration** dans la vue **Run** pour définir la connexion à un système Storm donné pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant un Job Storm **Talend**, consultez les sections décrivant comment créer et configurer un Job Storm **Talend** dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf lorsqu'explicitement indiqué, un scénario présente uniquement des Jobs de type **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données.

Scénarios associés

Pour des scénarios associés, consultez :

- *Scénario 2 : Mettre les données de sortie en mémoire tampon du serveur d'application Web*.
- *Scénario : Itérer une table de base de données et lister le nom des colonnes de la table*.
- *Scénario 1 : Filtrer une liste de noms à l'aide de conditions simples*.

tMemorizeRows

Propriétés du tMemorizeRows

Famille de composant	Misc	
Fonction	Le composant **tMemorizeRows** mémorise temporairement une collection de données entrantes en séquence ligne par ligne et instancie cette collection en indexant chaque ligne mémorisée à partir de 0. Le nombre maximum de lignes à mémoriser à un moment donné est défini dans la vue **Basic settings**.	
Objectif	Le **tMemorizeRows** mémorise une séquence de lignes passant dans ce composant et permet au(x) composant(s) suivant(s) d'effectuer des opérations sur les lignes mémorisées, selon vos besoins.	
	Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Row count to memorize	Saisissez le nom de lignes à définir.
	Columns to memorize	Sélectionnez les colonnes à mémoriser du schéma d'entrée.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant peut être utilisé en tant qu'étape intermédiaire dans un flux de données ou comme dernière étape avant de lancer un sous-job. Note: Vous pouvez utiliser la variable globale NB_LINE_ROWS afin de récupérer la valeur renseignée dans le champ **Row count to memorize** du composant **tMemorizeRows**.	
Connections	Liens de sortie (de composant à un autre) : **Row** : Main. **Trigger** : Run if, On Component Ok, On Component Error.	

	Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**NB_LINE_ROWS** : nombre de lignes mémorisées. Cette variable est une variable **After** et retourne un nombre entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

Scénario : Compter le nombre d'occurrences de différents âges

Ce scénario a pour objectif de compter le nombre d'âges différents se trouvant dans un groupe de 12 clients. Dans ce scénario, les données client sont générées de manière aléatoire.

Ce Job utilise les cinq composants suivants :

- **tRowGenerator** : ce composant génère 12 lignes de données client contenant les IDs, noms et âges des 12 clients.
- **tSortRow** : ce composant trie les 12 lignes selon les données de l'âge.
- **tMemorizeRows** : il mémorise temporairement un nombre spécifique de lignes de données entrantes à un moment donné et indexe les lignes de données mémorisées.
- **tJavaFlex** : ce composant compare la valeur des âges dans les données mémorisées par le composant précédent, compte le nombre d'occurrences des différents âges et affiche ces âges dans la vue **Run**.
- **tJava** : il affiche le nombre d'occurrences des différents âges.

Pour reproduire le scénario, procédez comme dans les sections suivantes.

Déposer et relier les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tRowGenerator**, **tSortRow**, **tMemorizeRows**, **tJavaFlex** et **tJava**.

2. Reliez le **tRowGenerator** au **tSortRow**, à l'aide d'un lien **Row** > **Main**.

3. Répétez l'opération pour connecter le **tSortRow**, le **tMemorizeRows** et le **tJavaFlex** à l'aide d'un lien **Row** > **Main**.

4. Connectez le **tRowGenerator** au **tJava** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Configurer les composants

Configurer le composant tRowGenerator

1. Double-cliquez sur le composant **tRowGenerator** afin d'ouvrir son éditeur.

2. Dans cet éditeur, cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes, que vous nommerez : *id*, *name*, *age*.

3. Dans la colonne **Type**, sélectionnez **Integer** pour les colonnes *id* et *age*.

4. Dans la colonne **Length**, saisissez *50* pour la colonne *name*.

5. Dans la colonne **Functions**, sélectionnez **random** pour *id* et *age*, puis sélectionnez **getFirstName** pour *name*.

6. Dans le champ **Number of Rows for RowGenerator**, saisissez *12*.

7. Dans la colonne **Column**, cliquez sur *age* afin d'ouvrir sa vue **Function parameters** dans la partie inférieure de l'éditeur.

Dans la colonne **Value** de la vue **Function parameters**, saisissez l'âge minimum et l'âge maximum à générer pour les 12 clients. Dans cet exemple, saisissez respectivement *10* et *25*.

Configurer le composant tSortRow

1. Double-cliquez sur le **tSortRow** afin d'ouvrir sa vue **Component**.

2. Dans le tableau **Criteria**, cliquez sur le bouton **[+]** afin d'ajouter une ligne.

3. Dans la colonne **Schema column**, sélectionnez la colonne de données sur laquelle vous souhaitez baser le tri. Dans cet exemple, sélectionnez *age*, puisque les âges seront comparés et comptés.

4. Dans la colonne **Sort num or alpha**, sélectionnez le type de tri. Dans cet exemple, comme *age* est un entier (integer), sélectionnez **num**, pour numérique.

5. Dans la colonne **Order asc or desc**, sélectionnez **desc** comme ordre de tri.

Configurer le composant tMemorizeRows

1. Double-cliquez sur le composant **tMemorizeRows** afin d'ouvrir sa vue **Component**.

2. Dans le champ **Row count to memorize**, saisissez le nombre maximum de lignes à mémoriser à un moment donné. Puisque vous devez comparer les âges de deux clients à chaque fois, saisissez *2*.

   Ce composant mémorise deux lignes au maximum à un moment donné et indexe toujours la nouvelle ligne entrante comme 0 et la ligne précédente comme 1.

3. Dans la colonne **Memorize** du tableau **Columns to memorize**, cochez la (les) case(s) de la (des) colonne(s) à mémoriser. Dans cet exemple, cochez la case correspondant à la colonne *age*.

Configurer les composants tJavaFlex et tJava

1. Double-cliquez sur le **tJavaFlex** afin d'ouvrir sa vue **Component**.

2. Dans la zone **Start code**, saisissez le code Java qui sera appelé durant la phase d'initialisation. Dans cet exemple, saisissez `int count=0;` afin de déclarer une variable *count* et de lui assigner la valeur *0*.

3. Dans la zone **Main code**, saisissez le code Java à appliquer à chaque ligne du flux de données. Dans ce scénario, saisissez :

```
if(!age_tMemorizeRows_1[0].equals(age_tMemorizeRows_1[1]))
{
count++;
}
System.out.println(age_tMemorizeRows_1[0]);
```

Ce code compare les deux âges mémorisés à chaque fois par le **tMemorizeRows** et compte un changement à chaque fois que des âges différents sont trouvés. Ce code affiche également les âges indexés comme 0 par le **tMemorizeRows**.

4. Dans la zone **End code**, saisissez le code qui sera appelé durant la phase de fermeture. Dans cet exemple, saisissez `globalMap.put("count", count);` afin d'écrire le résultat du compte.

5. Double-cliquez sur le **tJava** afin d'ouvrir sa vue **Component**.

6. Dans la zone **Code**, saisissez le code `System.out.println("Different ages: "+globalMap.get("count"));` afin de récupérer le résultat du compte.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** afin d'exécuter le Job, ou cliquez sur le bouton **Run** de la vue **Run**.

Dans la console, vous pouvez constater qu'il y a dix âges différents dans le groupe des 12 clients.

tMsgBox

Propriétés du tMsgBox

Famille de composant	Misc	
Fonction	Le composant **tMsgBox** ouvre une boîte de dialogue contenant un bouton **OK**, nécessitant une action de la part de l'utilisateur.	
Objectif	Ce composant est une pause graphique dans le cours du traitement.	
Basic settings	*Title*	Le texte saisi s'affiche dans la barre de titre de la boîte de dialogue.
	Buttons	Liste des boutons que vous souhaitez inclure dans la boîte de dialogue. Les combinaisons de bouton sont restreintes et ne peuvent être changées. Le bouton **Question** fait apparaître la case **Masquer la réponse**, qui une fois cochée vous permet de masquer la réponse que vous saisissez dans la fenêtre pop-up qui s'affiche quand vous exécutez votre Job.
	Icon	Icône de la barre de titre de la boîte de dialogue.
	Message	Texte libre à afficher dans la boîte de dialogue. Le texte peut être dynamique (ex : reprendre un nom de fichier).
Global Variables	**RESULT** : valeur de retour du composant. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé comme étape intermédiaire du traitement ou comme composant de début ou de fin de Job. Il peut être connecté à l'aide d'un lien **Row** ou **Iterate**.	

Scénario : Test de type ‘Hello world!'

Le scénario suivant représente un Job à un seul composant utilisé pour afficher l'heure et la date à la place du message "Hello World!".

1. Cliquez et déposez un composant **tMsgBox** dans l'espace de modélisation.

2. Paramétrez les propriétés d'affichage du message :

3. Le champ **Title** représente le titre de la boîte de message, vous pouvez utiliser n'importe quelle variable.

4. Dans le champ **Message**, saisissez le texte `"Current date is: "` entre guillemets doubles concaténé à l'aide d'un +, puis appuyez sur **Ctrl+Espace** afin d'afficher la liste d'auto-complétion, et sélectionnez la routine système `TalendDate.getCurrentDate`. Entourez-la de parenthèses.

5. Cliquez sur l'onglet **Run** puis exécutez le Job.

Le message affiche le texte défini précédemment et requiert une action de l'utilisateur pour disparaître et passer au composant suivant ou terminer le Job.

Après avoir cliqué sur le bouton **OK**, le log de la vue **Run** est mis à jour.

Voir également le *Guide utilisateur du Studio Talend*.

tRowGenerator

Propriétés du tRowGenerator

Famille de composant	Misc	
Fonction	Le composant **tRowGenerator** génère autant de lignes et de champs que vous souhaitez alimenter par des valeurs prises de façon aléatoire dans une liste définie.	
Objectif	Ce composant peut être utilisé à des fins de test, pour créer un flux d'entrée dans le cadre de jeux de tests.	
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant uniquement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
Basic settings	*RowGenerator editor*	L'éditeur vous permet de définir précisément les colonnes et la nature des données à générer. Vous pouvez prédéfinir des routines ou saisir vous-même la fonction à utiliser pour générer les données souhaitées. Notez que, dans un Job Storm, la valeur *-1* dans le champ **Number of rows for RowGenerator** du **RowGenerator editor** signifie la génération infinie de lignes de données d'entrée.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	La simplicité d'utilisation de l'éditeur du **tRowGenerator** vous permet de générer des données de façon aléatoire sans connaissance du langage Java.
Utilisation dans les Jobs Map/Reduce	Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme composant de début et requiert un composant de transformation comme lien de sortie. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Utilisation dans des Jobs Storm	Dans un Job Storm **Talend**, ce composant est utilisé comme composant de début. Les autres composants utilisés avec celui-ci doivent également être des composants Storm. Ils génèrent nativement du code Storm pouvant être exécuté directement dans un système Storm. La version de Storm ne supporte pas l'utilisation des variables globales. Vous devez utiliser l'onglet **Storm Configuration** dans la vue **Run** pour définir la connexion à un système Storm donné pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant un Job Storm **Talend**, consultez les sections décrivant comment créer et configurer un Job Storm **Talend** dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf lorsqu'explicitement indiqué, un scénario présente uniquement des Jobs de type **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

L'éditeur du **tRowGenerator** s'ouvre dans une fenêtre séparée constituée de deux parties :

- dans la partie supérieure de la fenêtre, une vue de définition de **Schema**,
- dans la partie inférieure, une vue **Function** de définition et d'aperçu des fonctions.

Définition du schéma

Pour commencer, vous devez définir la structure des données à générer.

- Ajoutez autant de colonnes que nécessaire à l'aide du bouton **plus [+]**.
- Dans la zone **Columns**, saisissez les noms des colonnes à créer et cochez la case **Key** si nécessaire.
- Assurez-vous d'avoir correctement défini la nature des données contenues dans la colonne en sélectionnant le type dans la colonne **Type**. En fonction du type sélectionné, la liste des fonctions proposées dans la colonne **Function** diffère. Cette information est obligatoire.

- D'autres informations, bien que facultatives, peuvent être utiles, telles que les colonnes **Length**, **Precision** ou **Comment**. Mais vous avez la possibilité de cacher les colonnes que vous n'utilisez pas, en cliquant sur le bouton **Columns** dans la barre d'outils et en décochant les colonnes que vous souhaitez cacher.
- Dans la zone **Function**, sélectionnez la routine/fonction prédéfinie si l'une des propositions correspond à vos besoins. Vous pouvez également ajouter à cette liste toute routine que vous avez développée et stockée dans la partie **Routine** du **Repository**. Vous avez enfin la possibilité de saisir directement la fonction que vous souhaitez utiliser dans la vue **Function**. Voir également : *Définition d'une fonction*.
- Cliquez sur **Refresh** pour visualiser un aperçu des données générées.
- Saisissez le nombre de lignes à générer. Plus le nombre de lignes est élevé, plus la durée d'exécution sera longue pour cette opération de génération.

Définition d'une fonction

Vous avez sélectionné les trois points **[...]** dans la colonne **Function** de l'éditeur du Schéma, car vous souhaitez personnaliser la fonction de génération.

- Sélectionnez l'onglet **Function parameters**.
- La zone **Parameter** affiche **Customized parameter** comme nom du paramètre (en lecture seule).

- Dans la zone **Value**, saisissez la fonction Java que vous souhaitez utiliser pour générer des données.
- Cliquez sur l'onglet **Preview** et cliquez sur le bouton **Preview** pour visualiser un échantillon des données qui seront générées.

Scénario : Génération aléatoire de fichiers de test

Le scénario suivant décrit un Job de deux composants, générant 50 lignes composées de la manière suivante : un ID aléatoire compris entre 1 et 3, un nom et un prénom pris de manière aléatoire dans une liste de chaînes de caractères, et le champ **Date** définie dans une période donnée.

- Cliquez et déposez un composant **tRowGenerator** et un composant **tLogRow** dans l'espace de modélisation.
- Cliquez-droit sur le composant **tRowGenerator** et sélectionnez le lien **Row** > **Main**. Faites glisser ce lien jusqu'au composant **tLogRow**.
- Double-cliquez sur le composant **tRowGenerator** pour ouvrir l'éditeur.
- Définissez les champs à générer.

- La colonne ID (*random_ID*) est de type entier (**integer**), les noms et prénoms sont de type chaîne de caractères (**String**) et la *Date* est de type **Date**.
- Dans la colonne **Function**, sélectionnez la fonction adaptée dans la liste ou les trois points **[...]** pour la fonction personnalisée.
- Dans l'onglet **Function parameters**, définissez les valeurs à prendre de façon aléatoire.

Parameter	Value	Comment
customize parameter	1,2,3	

- Les colonnes *First_Name* et *Last_Name* sont à générer à l'aide d'une fonction `getAsciiRandomString` prédéfinie dans les routines système. Par défaut, la longueur définie est de 6 caractères, mais vous pouvez la changer.
- La colonne *Date* appelle également une fonction prédéfinie `getRandomDate`. Vous pouvez éditer les valeurs de paramètre dans l'onglet **Function parameters**.
- Dans le champ **Number of Rows**, paramétrez le nombre de lignes à *50*.
- Cliquez sur **OK** pour valider les paramètres.
- Double-cliquez sur le composant **tLogRow** pour visualiser les paramètres dans l'onglet **Basic settings**. Retenez les paramètres par défaut.
- Appuyez sur **F6** pour exécuter le Job.

```
Starting job JavaRowGenerate at 11:44 10/04/2007.
Running process with context: Default
1|iH6mtj|y3hSXH|Wed Apr 16 11:44:15 CEST 2008
3|cityeQ|uvnKkO|Sun Jan 06 11:44:16 CET 2008
2|X2O0DP|1SVzKT|Wed Oct 26 11:44:16 CEST 2005
1|DSLuE1|S8u15i|Sat Mar 04 11:44:16 CET 2006
3|cc4znX|yuc9cf|Thu Jan 20 11:44:16 CET 2005
3|NwK3PN|lnNyDU|Mon Jun 06 11:44:16 CEST 2005
3|CtO6Ba|pCQgwp|Sat Aug 07 11:44:16 CEST 2004
3|XMt7KN|SUIzFn|Sun May 09 11:44:16 CEST 2004
1|jiPor7|l45rp7|Wed Aug 25 11:44:16 CEST 2004
1|OBN4TX|hywNdP|Sat Feb 09 11:44:16 CET 2008
2|kbVUuE|s21FA0|Tue Jan 27 11:44:16 CET 2004
2|zQlteU|VNHb5w|Sun Oct 05 11:44:16 CEST 2008
3|4nrniu|5Tbnxd|Fri Nov 14 11:44:16 CET 2008
2|7A0qqu|s3jEzJ|Tue Feb 13 11:44:16 CET 2007
3|FgVoTg|uYhJmF|Thu Mar 01 11:44:16 CET 2007
1|Sp4e9P|zXcTK5|Thu Apr 08 11:44:16 CEST 2004
1|E3A4Q3|vKNocj|Wed Jan 11 11:44:16 CET 2006
1|pOMGpG|koklW2|Thu Apr 22 11:44:16 CEST 2004
3|9XOgm8|CCbFya|Sat Feb 18 11:44:16 CET 2006
2|itZXWx|LhQBx1|Tue Apr 18 11:44:16 CEST 2006
1|On1dwv|E7Yaqn|Fri Dec 10 11:44:16 CET 2004
1|UPTZ9M|8s902c|Fri Nov 11 11:44:16 CET 2005
1|WhIT0u|KWn8hO|Mon May 07 11:44:16 CEST 2007
1|LbbeFy|Eyn1lc|Thu Mar 16 11:44:16 CET 2006
```

Les 50 lignes sont générées selon les paramètres définis dans l'éditeur du composant **tRowGenerator** et sont affichées en sortie standard de la vue **Run**.

tServerAlive

Propriétés du tServerAlive

Famille de composant	Misc	
Fonction	Le **tServerAlive** valide le statut de la connexion à un hôte défini.	
Objectif	Le **tServerAlive** permet de vérifier si la connexion à l'hôte défini est active ou non. Ceci est indiqué par la valeur booléenne retournée par la variable globale *SERVER_ALIVE_RESULT*.	
Basic settings	*Protocol*	Dans la liste de protocoles, sélectionnez **Ping** ou **Telnet**.
	Host	Adresse IP ou nom d'hôte du serveur distant à vérifier.
	Timeout Interval (in seconds)	Définissez le délai avant suspension en secondes. Si l'hôte cible est atteint pendant l'intervalle défini, la valeur de la variable globale *SERVER_ALIVE_RESULT* est *true*. Sinon, sa valeur est *false*.
	Port	Numéro de port d'écoute du serveur Telnet. Cette option est disponible quand **Telnet** est sélectionné dans la liste **Protocol**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case afin de collecter les données de log au niveau du Job ainsi qu'au niveau des composants.
Global Variables	**SERVER_ALIVE_RESULT** : statut de la connexion à un hôte distant. Cette variable est une variable **After** et retourne un booléen. **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé en tant que sous-job à un seul composant.	
Limitation		

Scénario : Valider le statut de la connexion à un hôte distant

Dans ce scénario, le composant **tServerAlive** est utilisé afin d'envoyer une requête ping à un hôte distant. La valeur de sa variable globale *SERVER_ALIVE_RESULT* est extraite afin de connaître le statut de la connexion.

Déposer et relier les composants

1. Déposez les composants suivants depuis la **Palette** dans l'espace de modélisation graphique : un **tServerAlive**, un **tFixedFlowInput** et un **tLogRow**.

2. Reliez le **tServerAlive** au **tFixedFlowInput** à l'aide d'un lien **Trigger** > **On Subjob Ok**.
3. Reliez le **tFixedFlowInput** au **tLogRow** à l'aide d'un lien **Row > Main**.

Configurer les composants

1. Double-cliquez sur le **tServerAlive** afin d'ouvrir sa vue **Basic settings**.

2. Dans la liste **Protocol**, sélectionnez **Ping**.
3. Dans le champ **Host**, saisissez l'adresse IP du serveur distant auquel envoyer une requête ping.
4. Dans le champ **Timeout Interval**, saisissez le délai avant suspension en secondes.
5. Double-cliquez sur le **tFixedFlowInput** afin d'ouvrir sa vue **Basic settings**.

6. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir l'éditeur de schéma.

7. Cliquez sur le bouton **[+]** afin d'ajouter une colonne de type booléen nommée *status*.

   Cliquez sur **OK** afin de valider les modifications et fermer l'éditeur.

   Dans la boîte de dialogue qui s'affiche, cliquez sur **Yes** afin de propager les changements aux autres composants.

8. Dans le champ **Value** de la table **Values**, appuyez sur **Ctrl + Espace** et sélectionnez *tServerAlive_1_SERVER_ALIVE_RESULT* dans la liste des variables.

9. Double-cliquez sur le **tLogRow** afin d'ouvrir sa vue **Basic settings**.

10. Sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Sauvegarder et exécuter le Job

1. Appuyez sur **Ctrl+S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** afin d'exécuter le Job.

```
[statistics] connecting to socket on port 3546
[statistics] connected
.------.
|tLogRow_1|
|=-----=|
|status|
|=-----=|
|true  |
'------'
```

Comme montré par la capture d'écran, le serveur distant est actif.

Composants Orchestration

Ce chapitre passe en revue les propriétés des principaux composants présents dans la famille **Orchestration** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **Orchestration** regroupe les composants vous permettant de séquencer et d'orchestrer les tâches ou les traitements de vos Jobs et sous-Jobs.

tFileList

Le **tFileList** appartient à deux familles : File et Orchestration. Pour plus d'informations concernant le composant **tFileList**, consultez *tFileList*.

tFlowToIterate

Propriétés du tFlowToIterate

Famille du composant	Orchestration	
Fonction	Le composant **tFlowToIterate** effectue une itération sur les données d'entrée et génère des variables globales.	
Objectif	Le **tFlowToIterate** est utilisé afin de lire des données ligne par ligne et de stocker les entrées des données dans des variables globales itératives.	
Basic settings	*Use the default (key, value) in global variables*	Lorsque cette case est cochée, le système utilise la valeur par défaut de la variable globale utilisée dans le Job.
	Customize	**key** : Entrez le nom de la nouvelle variable globale. Appuyez sur **Ctrl+Espace** pour accéder à l'ensemble des variables disponibles, qu'elles soient globales ou définies par l'utilisateur.
		value : Cliquez dans la cellule pour accéder à la liste des colonnes rattachées à la variable globale définie.
Utilisation	Le **tFlowToIterate** n'est pas un composant de début, et il requiert un composant de sortie.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **CURRENT_ITERATION** : numéro de séquence de l'itération courante. Cette variable est une variable **Flow** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Iterate. **Trigger** : Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Transformer un flux de données en liste

Le scénario suivant décrit un Job permettant de lire une liste de fichiers à partir d'un fichier d'entrée donné. Il effectue également une itération sur chacun des fichiers et affiche leur contenu ligne par ligne dans la console de log **Run**.

Construire le Job

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l'espace de modélisation : deux **tFileInputDelimited**, un **tFlowToIterate** et un **tLogRow**.

2. Cliquez-droit sur chacun des composants pour les relier entre eux. Connectez le premier **tFileInputDelimited** au **tFlowToIterate** à l'aide d'un lien de type **Row Main**, puis le **tFlowToIterate** au deuxième **tFileInputDelimited** à l'aide d'un lien de type **Iterate**, enfin le second **tFileInputDelimited** au **tLogRow** à l'aide d'un lien de type **Row Main**.

Configurer les composants

1. Dans l'espace de modélisation graphique, double-cliquez sur le premier **tFileInputDelimited** pour afficher sa vue **Component**.

2. Dans l'onglet **Basic settings**, cliquez sur le bouton **[...]** situé à côté du champ **File Name** pour déterminer le chemin d'accès au fichier d'entrée.

Le champ **File Name** est obligatoire.

Le fichier d'entrée utilisé dans ce scénario est *Customers*. Il s'agit d'un fichier texte contenant trois autres fichiers texte simples : *Name*, *E-mail* et *Address*. Le fichier *Name* est constitué d'une colonne contenant le nom des clients et le fichier *E-mail* d'une colonne contenant l'adresse email de ces clients. Le troisième fichier texte, *Address*, est composé d'une colonne comprenant l'adresse postale des clients.

Renseignez les autres champs en fonction de votre fichier. Pour plus d'informations sur le **tFileInputDelimited**, consultez *Propriétés du tFileInputDelimited*. Dans ce scénario, l'en-tête, le pied de page et le nombre limite de lignes à traiter ne sont pas définis. Donc, laissez les champs **Header** (en-tête) et **Footer** (pied de page) et **Limit** par défaut.

3. Cliquez sur le bouton **Edit schema** pour décrire la structure des données du fichier d'entrée. Dans ce scénario, le schéma comporte une seule colonne appelée *FileName*.

4. Dans l'espace de modélisation graphique, sélectionnez le **tFlowToIterate** et cliquez sur l'onglet **Component** pour définir sa configuration de base (**Basic settings**).

Cliquez sur le bouton **[+]** pour ajouter des lignes de paramètres et définissez vos variables. Cliquez dans la cellule **key** afin de saisir le nom de la variable souhaité. Dans ce scénario, la variable est la suivante : `"Name_of_File"`.

Vous pouvez également cocher la case **Use the default (key, value) in global variables** pour utiliser la valeur par défaut des variables globales.

5. Dans l'espace de modélisation graphique, double-cliquez sur le second composant **tFileInputDelimited** pour ouvrir sa vue **Component**.

Dans le champ **File Name**, renseignez le répertoire des fichiers à lire puis appuyez sur les touches **Ctrl +Espace** pour sélectionner la variable globale *"Name_of_File"*. Dans ce scénario, la syntaxe est la suivante :

```
"C:/scenario/flow_to_iterate/"+((String)globalMap.get("Name_of_File"))
```

Cliquez sur **Edit schema** afin de définir le nom de la colonne du schéma. Nommez-la, dans ce scénario, *RowContent*.

Renseignez les autres champs selon vos besoins. Pour plus d'informations, consultez *Propriétés du tFileInputDelimited*.

6. Dans l'espace de modélisation graphique, double-cliquez sur le dernier composant, le **tLogRow**, pour afficher sa vue **Component** et définir sa configuration de base (**Basic settings**).

Définissez les paramètres selon vos besoins. Pour plus d'informations, consultez *Propriétés du tLogRow*.

Sauvegarder et exécuter le Job

1. Enregistrez votre Job en appuyant sur les touches **Ctrl+S**.

2. Appuyez sur **F6** pour l'exécuter ou cliquez sur le bouton **Run** dans la vue **Run**.

```
Starting job Practice_8_27 at 16:12 28/08/2012.

[statistics] connecting to socket on port 3580
[statistics] connected
RowContent: Madison Moore
RowContent: Andrew Taylor
RowContent: Christopher Anderson
RowContent: madison_moor@hotmail.com
RowContent: andre_taylor@usamail.com
RowContent: chrisopher_anderson@hotmail.com
RowContent: Madison Moore
RowContent:     100 MAIN ST
RowContent:     PO BOX 1022
RowContent:     SEATTLE WA 98104
RowContent:     USA
RowContent: Andrew Taylor
RowContent:     300 BOYLSTON AVE E
RowContent:     SEATTLE WA 98102
RowContent:     USA
RowContent: Christopher Anderson
RowContent:     500 DOCTOR ST
RowContent:     SEATTLE WA 98100
RowContent:     USA
[statistics] disconnected
Job Practice_8_27 ended at 16:12 28/08/2012. [exit code=0]
```

La console affiche, pour chaque ligne, le nom du client, son e-mail et son adresse, précédés du nom de la colonne du schéma.

tForeach

Propriétés du tForeach

Famille de composant	Orchestration	
Fonction	Le composant **tForeach** crée une boucle sur une liste pour un lien **Iterate**.	
Objectif	Le **tForeach** vous permet de créer une boucle sur une liste pour un lien **Iterate**.	
Basic settings	*Values*	Utilisez le bouton **[+]** pour ajouter des lignes au tableau **Values**. Cliquez ensuite sur les champs afin de saisir, entre guillemets, les valeurs de la liste sur laquelle effectuer une boucle.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **CURRENT_VALUE** : valeur sur laquelle s'effectue l'itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le **tForeach** est un composant d'entrée. Un lien **Iterate** est requis pour relier ce composant à un autre.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Faire une boucle sur une liste et récupérer les valeurs

Ce scénario décrit un Job à deux composants dans lequel une liste est créée et sur laquelle est effectuée une boucle, via un composant **tForeach**. Les valeurs sont récupérées par un composant **tJava**.

1. Déposez de la **Palette** dans l'espace de modélisation graphique les composants **tForeach** et **tJava**.

2. Reliez le **tForeach** au **tJava** à l'aide d'un lien **Row > Iterate**.

3. Double-cliquez sur le **tForeach** afin d'ouvrir sa vue **Basic settings**.

4. Cliquez sur le bouton pour ajouter autant de lignes que nécessaire dans le tableau **Values**.

5. Cliquez sur les champs **Value** afin de saisir les valeurs de la liste, entre guillemets doubles.

6. Double-cliquez sur le **tJava** afin d'ouvrir sa vue **Basic settings**.

7. Saisissez le code Java suivant dans la zone **Code** : `System.out.println(globalMap.get("tForeach_1_CURRENT_VALUE")+"_out");`

8. Sauvegardez le Job et appuyez sur **F6** pour l'exécuter.

   La console affiche la liste des valeurs récupérées du **tForeach**, chaque valeur ayant le suffixe _out.

Job tForEach_2
Basic Run
Debug Run
Advanced Settings
Target Exec
Execution
Run
Kill
Clear
Starting job tForEach_2 at 14:28 04/10/2010.
[statistics] connecting to socket on port 3409
[statistics] connected
element_1_out
element_2_out
element_3_out
element_4_out
[statistics] disconnected
Job tForEach_2 ended at 14:28 04/10/2010. [exit code=0]
Line limit
100

tInfiniteLoop

Propriétés du tInfiniteLoop

Famille de composant	Orchestration	
Fonction	Le composant **tInfiniteLoop** fait une boucle à l'infini sur l'exécution d'une tâche.	
Objectif	Le **tInfiniteLoop** permet d'exécuter automatiquement une tâche ou un Job dans une boucle, à l'infini.	
Basic settings	*Wait at each iteration (in milliseconds)*	Saisissez le temps, en millisecondes, d'attente entre chaque itération.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Le composant **tInifniteLoop** est un composant de début et requiert une connexion de type Iterate au composant suivant.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **CURRENT_ITERATION** : numéro de séquence de l'itération courante. Cette variable est une variable **Flow** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.	

Scénario associé

Pour un exemple d'utilisation proche, consultez *Scénario : Exécuter un Job à partir d'une boucle* du composant **tLoop**.

tIterateToFlow

Propriétés du tIterateToFlow

Famille de composant	Orchestration	
Fonction	Le composant **tIterateToFlow** transforme une liste en flux de données pouvant être traité.	
Objectif	Ce composant permet de transformer des données non traitables en flux traitable.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant uniquement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Mapping	**Column** :Saisissez le nom des colonnes à créer. **Value** : Appuyez sur **Ctrl+Espace** pour accéder à toutes les variables disponibles soit globales soit celles que vous avez définies.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant n'est pas un composant de début (fond vert) et requiert un composant de sortie.	
Connections	Liens de sortie (de ce composant à un autre) :	

	Row : Main. **Trigger** : Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.

Scénario : Transformer une liste de fichiers en flux de données

Le scénario suivant décrit un Job permettant de faire une boucle sur une liste de fichiers, de récupérer leur nom, ainsi que la date en cours, pour ensuite transformer ces données en un flux qui sera affiché dans la console.

- Cliquez et déposez les composants suivants : **tFileList**, **tIterateToFlow** et **tLogRow**.
- Connectez le **tFileList** au **tIterateToFlow** à l'aide d'un lien **Iterate** et connectez le Job au composant **tLogRow** à l'aide d'un lien **Row Main**.
- Dans la vue **Component** du composant **tFileList**, donnez le chemin d'accès au répertoire contenant la liste de fichiers.

- Dans cet exemple, les fichiers sont trois fichiers .txt simples stockés dans le répertoire *Countries*.
- Dans la liste **Case sensitive**, sélectionnez **No** (Non) pour ne pas tenir compte de la casse.
- Sélectionnez le composant **tIterateToFlow** et cliquez sur **Edit Schema** pour modifier le nouveau schéma.

- Ajoutez deux nouvelles colonnes : *Filename* de type **String** et *Date* de type **date**. Assurez-vous d'avoir paramétré le bon modèle en Java.
- Cliquez sur **OK** pour valider.
- Une boîte de dialogue s'ouvre et vous demande si vous souhaitez propager les changements. Cliquez sur **Yes** (Oui).
- Notez que ce nouveau schéma apparaît dans le tableau **Mapping**.

- Dans chaque cellule de la colonne **Value**, appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales et à celles que vous avez définies.
- Pour la colonne *Filename*, utilisez la variable globale : `tFileList_1CURRENT_FILEPATH`. Elle reprend le chemin d'accès courant pour recueillir le nom de chaque fichier, le Job fait une boucle sur ce répertoire.
- Pour la colonne *Date*, utilisez la routine Talend : `TalendDate.getCurrentDate()`.
- Dans la vue **Component** du composant **tLogRow**, cochez la case **Print values in cells of a table**.
- Enregistrez votre Job et exécutez-le, en appuyant sur **F6**.

```
Starting job tIterateToFlow at 10:28 10/03/2010.

[statistics] connecting to socket on port 3773
[statistics] connected
.----------------------------+-------------------.
|                   tLogRow_1                    |
|=---------------------------+-------------------=|
|Filename                    |Date               |
|=---------------------------+-------------------=|
|D:\Input\Countries\in-01.txt|2010-03-10 10:28:56|
|D:\Input\Countries\in-02.txt|2010-03-10 10:28:56|
|D:\Input\Countries\in-03.txt|2010-03-10 10:28:56|
'----------------------------+-------------------'

[statistics] disconnected
Job tIterateToFlow ended at 10:28 10/03/2010. [exit code=0]
```

Le chemin d'accès apparaît dans la colonne *Filename* et la date en cours apparaît dans la colonne *Date*.

tLoop

Propriétés du tLoop

Famille de composant	Orchestration	
Fonction	Le composant **tLoop** fait une boucle sur l'exécution d'une tâche.	
Objectif	Le **tLoop** permet d'exécuter automatiquement une tâche ou un Job dans une boucle.	
Basic settings	*Loop Type*	Sélectionnez le type de boucle à effectuer : **For** ou **While**. **For** : La tâche ou le Job est exécuté(e) pour le nombre de boucle défini. **While** : La tâche ou le Job est exécuté(e) jusqu'à ce que la condition soit rencontrée.
For	*From*	Saisissez le numéro de la première instance sur laquelle la boucle doit commencer. Si vous sélectionnez l'instance numéro 2 comme instance de départ et que vous prenez un écart de 2, cela signifie que la boucle se fera sur tous les nombres pairs.
	To	Saisissez la dernière instance sur laquelle la boucle doit passer.
	Step	Saisissez l'écart entre deux instances avec lequel la boucle doit s'effectuer. Un écart de 2 signifie que la boucle passe toutes les deux instances.
	Values are increasing	Cochez cette case pour autoriser uniquement une séquence croissante.
While	*Declaration*	Saisissez une expression déclenchant la boucle.
	Condition	Saisissez la condition devant être rencontrée pour que la boucle s'arrête.
	Iteration	Saisissez l'expression décrivant l'opération à effectuer à chaque boucle.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Le composant **tLoop** est un composant de début et requiert une connexion de type **Iterate** au composant suivant.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **CURRENT_VALUE** : valeur courante. Disponible uniquement pour les boucles de type **For**. Cette variable est une variable **Flow** et retourne un entier. **CURRENT_ITERATION** : numéro de séquence de l'itération courante. Cette variable est une variable **Flow** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) :	

	Row : Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Exécuter un Job à partir d'une boucle

Ce scénario décrit un Job constitué d'un Job parent et d'un Job enfant. Le Job parent exécute une boucle qui permet d'exécuter un Job enfant cinq fois, avec une pause de trois secondes entre chaque exécution.

1. Créez un Job nommé *Parent* puis déposez un **tLoop**, un **tRunJob** et un **tSleep** dans l'espace de modélisation graphique.

2. Reliez le **tLoop** au **tRunJob** l'aide d'un lien **Row** > **Iterate** puis reliez le **tRunJob** au **tSleep** à l'aide d'un lien **Row** > **Main**.

3. Créez un Job nommé *Child* puis déposez un **tRowGenerator** et un **tLogRow** dans l'espace de modélisation graphique.

4. Reliez le **tRowGenerator** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

5. Dans la vue **Basic settings** du composant **tLoop**, définissez le type de boucle (**For**) puis saisissez le numéro de l'instance sur lequel commencer (*1*), le numéro de l'instance sur lequel finir (*5*) et l'écart à utiliser (*1*).

6. Dans la vue **Basic settings** du composant **tRunJob**, sélectionnez le Job *Child* dans le champ **Job**.

7. Dans la vue **Basic settings** du composant **tSleep**, saisissez la durée de la pause en secondes dans le champ **Pause**. Pour cet exemple, la pause est de *3* secondes.

8. Double-cliquez sur le composant **tRowGenerator** afin d'ouvrir l'éditeur de schéma. Cliquez ensuite sur le bouton **[+]** afin d'ajouter quatre colonnes :

   - *id*, afin de générer les numéros de séquence
   - *firstname*, afin de générer des prénoms aléatoires
   - *lastname*, afin de générer des noms aléatoires
   - *city*, afin de générer des noms de ville aléatoires

   Pour plus d'informations concernant le **tRowGenerator**, consultez *tRowGenerator*.

9. Appuyez sur **F6** afin d'exécuter le Job *Parent*.

   Le Job *Child* est exécuté cinq fois avec une pause de trois secondes entre chacune des deux exécutions. Comme configuré dans le composant **tRowGenerator**, des informations personnelles aléatoires sont affichées dans la console **Run**.

tPostjob

Propriétés du tPostjob

Famille du composant	Orchestration	
Fonction	Le composant **tPostjob** démarre l'exécution d'un post-job.	
Objectif	Le **tPostjob** déclenche une action requise après l'exécution d'un Job.	
Utilisation	Le **tPostjob** est un composant de début. Il ne peut être relié au composant suivant qu'avec une connexion de type **Iterate**.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Trigger** : OnComponentOk. Liens d'entrée (d'un autre composant à celui-ci) : **Trigger** : Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.	

Pour plus d'informations sur le composant **tPostjob**, consultez le *Guide utilisateur du Studio Talend*.

Scénario associé

Pour des scénarios utilisant le **tPostjob**, consultez *Scénario : Gérer des fichiers avant et après l'exécution d'un Job* et *Scénario : Créer un diagramme en barres à partir de données d'entrée* .

tPrejob

Propriétés du tPrejob

Famille du composant	Orchestration	
Fonction	Le composant **tPrejob** démarre l'exécution d'un pré-job.	
Objectif	Le **tPrejob** déclenche une action requise pour l'exécution d'un Job.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le **tPrejob** est un composant de début. Il ne peut être relié au composant suivant qu'avec une connexion de type **Iterate**.	
Connections	Liens de sortie (de ce composant à un autre) : **Trigger** : OnComponentOk. Liens d'entrée (d'un autre composant à celui-ci) : **Trigger** : Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.	

Pour plus d'informations sur le composant **tPrejob**, consultez le *Guide utilisateur du Studio Talend*.

Scénario : Gérer des fichiers avant et après l'exécution d'un Job

Dans ce scénario, un pré-Job et un post-Job sont ajoutés dans le Job décrit dans *Scénario 2 : Trouver des fichiers dupliqués entre deux dossiers* afin de gérer les fichiers avant et après l'exécution du Job principal.

Comme décrit dans le scénario mentionné précédemment, lorsque le Job principal démarre, il crée un fichier texte temporaire afin de stocker les noms des fichiers récupérés. Si le fichier temporaire spécifié existe déjà, il est écrasé.

Afin de prévenir toute perte de données, le pré-Job enregistre une copie de sauvegarde du fichier spécifié en le renommant avant que le Job principal s'exécute.

Après l'exécution du Job principal, le post-Job supprime le fichier temporaire créé par le Job et restaure le fichier de sauvegarde, s'il existe.

Ajouter et relier les composants

1. Ouvrez le Job principal dans l'espace de modélisation graphique puis ajoutez-y un **tPrejob**, un **tPostjob**, un **tFileDelete** et deux **tFileCopy**.
2. Reliez le **tPrejob** au premier **tFileCopy** à l'aide d'un lien **Trigger** > **On Component Ok** afin de créer le pré-Job.
3. Reliez le **tPostjob** au **tFileDelete** à l'aide d'un lien **Trigger** > **On Component Ok** et reliez l'autre **tFileDelete** à l'autre **tFileCopy** afin de créer le post-Job.
4. Nommez les composants afin d'identifier plus facilement leur rôle.

Configurer les composants

Configurer le pré-Job

Dans le pré-Job, le **tPrejob** n'a aucun paramètre ou propriété à définir. Il déclenche seulement le **tFileCopy** avant l'exécution du Job principal afin de renommer le fichier spécifié. Tous les paramètres nécessaires sont donc définis dans le **tFileCopy**.

1. Double-cliquez sur le **tFileCopy** afin d'ouvrir sa vue **Basic settings**.

2. Dans le champ **File Name**, saisissez le chemin ainsi que le nom du fichier temporaire à renommer, *D:/temp/tempdata.csv* dans ce scénario.

3. Dans le champ **Destination directory**, spécifiez le chemin ou parcourez votre système vers le dossier cible. Dans cet exemple, la copie de sauvegarde est enregistrée dans le même dossier, *D:/temp/*.

4. Cochez la case **Rename** et spécifiez le nouveau nom du fichier dans le champ **Destination filename**, *backup-tempdata.csv*. Laissez les autres paramètres tels qu'ils sont.

Configurez le post-Job

Dans le post-Job, le **tPostjob** n'a aucun paramètre ou propriété à définir. Il déclenche seulement le **tFileDelete** après l'exécution du Job principal afin de supprimer le fichier temporaire utilisé pour stocker les noms des fichier récupérés. Il déclenche ensuite le **tFileCopy** afin renommer le fichier de sauvegarde avec son nom d'origine.

1. Dans le champ **File Name** de la vue **Basic settings** du **tFileDelete**, saisissez le chemin et le nom du fichier temporaire à supprimer, *D:/temp/tempdata.csv* dans ce scénario. Laissez les autres champs tels qu'ils sont.

2. Double-cliquez sur le **tFileCopy** afin d'ouvrir sa vue **Basic settings**.

3. Dans le champ **File Name**, saisissez le chemin et le nom du fichier de sauvegarde, *D:/temp/backup-tempdata.csv* dans ce scénario.

4. Dans le champ **Destination directory**, spécifiez le chemin ou parcourez votre système vers le dossier cible, *D:/temp/* dans ce scénario.

5. Cochez la case **Rename** et spécifiez le nom original du fichier temporaire dans le champ **Destination filename**, *tempdata.csv*.

6. Cochez la case **Remove source file** afin de supprimer le fichier de sauvegarde après l'action de renommage. Laissez les autres champs tels qu'ils sont.

Résultat d'exécution attendu

Si le fichier temporaire spécifié existe, ses données sont conservées dans une copie de sauvegarde avant l'exécution du Job principal et restaurées par la suite.

Si le fichier temporaire n'existe pas, les deux **tFileCopy** génèrent une erreur mais n'empêchent pas l'exécution du Job principal.

Pour le résultat de l'exécution du Job principal, consultez *Scénario 2 : Trouver des fichiers dupliqués entre deux dossiers*.

Scénario associé

Pour un autre scénario utilisant le **tPrejob**, consultez *Scénario : Créer un diagramme en barres à partir de données d'entrée* .

tReplicate

Propriétés du tReplicate

Famille de composant	Orchestration	
Fonction	Le composant **tReplicate** duplique le schéma entrant en deux flux identiques. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job **Talend** Map/Reduce ou Storm. Dans ces deux cas, le **tReplicate** appartient à la famille de composant Processing.	
Objectif	Ce composant permet d'effectuer différentes opérations sur le même schéma. Si vous avez souscrit à l'une des solutions Big Data du *Studio Talend*, vous pouvez utiliser ce composant dans un Job **Talend** Map/Reduce afin de générer du code Map/Reduce. Dans ce cas, le **tReplicate** appartient à la famille de composants Processing.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.
		Built-in : Le schéma est créé et conservé pour ce composant uniquement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository: Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant n'est pas un composant de début (fond vert) puisqu'il requiert un composant d'entrée et un composant de sortie.
Utilisation dans des Jobs Map/ Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Utilisation dans des Jobs Storm	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez également utiliser ce composant en tant que composant Storm. Dans un Job Storm **Talend**, ce composant est utilisé comme étape intermédiaire. Les autres composants utilisés avec celui-ci doivent également être des composants Storm. Ils génèrent nativement du code Storm pouvant être exécuté directement dans un système Storm. La version de Storm ne supporte pas l'utilisation des variables globales. Vous devez utiliser l'onglet **Storm Configuration** dans la vue **Run** pour définir la connexion à un système Storm donné pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant un Job Storm **Talend**, consultez les sections décrivant comment créer et configurer un Job Storm **Talend** dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf lorsqu'explicitement indiqué, un scénario présente uniquement des Jobs de type **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main **Trigger** : Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à un celui-ci) : **Row** : Main, Reject. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Répliquer un flux et trier deux flux identiques

Ce scénario décrit un Job lisant un flux d'entrée contenant des noms et des états dans un fichier CSV, réplique le flux d'entrée, puis trie les deux flux identiques selon le nom et l'état respectivement et affiche les données triées dans la console.

Construire le Job

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tFileInputDelimited**, un **tReplicate**, deux **tSortRow** et deux **tLogRow**.

2. Reliez le **tFileInputDelimited** au **tReplicate** à l'aide d'un lien **Row** > **Main**.

3. Répétez les étapes ci-dessus afin de connecter le composant **tReplicate** aux deux **tSortRow** puis les deux **tSortRow** aux deux **tLogRow** (un chacun).

4. Renommez les composants afin de mieux identifier leur rôle au sein du Job.

Configurer les composants

1. Double-cliquez sur le **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **File name/Stream** afin de parcourir votre système jusqu'au fichier duquel vous souhaitez lire le flux d'entrée. Dans cet exemple, le fichier d'entrée se nomme *Names&States.csv* et contient deux colonnes : *name* et *state*.

```
name;state
Andrew Kennedy;Mississippi
Benjamin Carter;Louisiana
Benjamin Monroe;West Virginia
```

```
Bill Harrison;Tennessee
Calvin Grant;Virginia
Chester Harrison;Rhode Island
Chester Hoover;Kansas
Chester Kennedy;Maryland
Chester Polk;Indiana
Dwight Nixon;Nevada
Dwight Roosevelt;Mississippi
Franklin Grant;Nebraska
```

3. Renseignez les champs **Header**, **Footer** et **Limit** selon vos besoins. Dans cet exemple, saisissez *1* dans le champ **Header** pour ignorer la première ligne du fichier d'entrée.

4. Cliquez sur **Edit schema** pour définir la structure des données du flux d'entrée.

5. Double-cliquez sur le premier **tSortRow** pour ouvrir sa vue **Basic settings**.

6. Dans la table **Criteria**, cliquez sur le bouton **[+]** pour ajouter une ligne et configurez les paramètres de tri pour la colonne du schéma à traiter. Pour trier les données d'entrée selon le nom, sélectionnez *name* pour **Schema column**. Sélectionnez *alpha* comme type de tri et *asc* comme ordre de tri.

   Pour plus d'informations concernant ces paramètres, consultez *tSortRow*.

7. Double-cliquez sur le second **tSortRow** et répétez l'étape ci-dessus pour configurer les paramètres pour la colonne contenant les états.

8. Dans la vue **Basic settings** de chaque **tLogRow**, sélectionnez **Table** dans la zone **Mode** pour un meilleur affichage des résultats d'exécution du Job.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.

2. Exécutez le Job en appuyant sur **F6** ou en cliquant sur le bouton **Run** de la vue **Run**.

Les données triées par nom et les données triées par état sont affichées dans la console.

tRunJob

Le **tRunJob** appartient à deux familles : Orchestration et System. Pour plus d'informations concernant le composant **tRunJob**, consultez *tRunJob*.

tUnite

Propriétés du tUnite

Famille de composant	Orchestration	
Fonction	Le composant **tUnite** fusionne des données de diverses sources, basées sur un même schéma. *Le **tUnite** ne peut exister dans une boucle de flux de données. Par exemple, si un flux de données passe par plusieurs composants **tMap** afin de générer deux flux, ils ne peuvent être regroupés dans un **tUnite**.*	
Objectif	Ce composant centralise des données provenant de sources diverses et hétérogènes.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.
		Built-in : Le schéma est créé et conservé pour ce composant uniquement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant n'est pas un composant de début; il requiert un ou plusieurs composants en entrée et un composant de sortie.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Itération sur des fichiers et fusion de contenu

Le Job suivant effectue une itération sur une liste de fichiers puis fusionne leur contenu et affiche le contenu final des deux colonnes dans la console.

Déposer et relier les composants

1. Cliquez et déposez les composants suivants dans l'espace de modélisation : **tFileList**, **tFileInputDelimited**, **tUnite** et **tLogRow**.

2. Connectez le composant **tFileList** au **tFileInputDelimited** à l'aide d'un lien **Iterate** et connectez les autres composants à l'aide de liens **Row main**.

Configurer les composants

1. Dans l'onglet **Basic settings** du composant **tFileList**, sélectionnez le répertoire dans lequel les fichiers à fusionner sont stockés.

Les fichiers sont très basiques et contiennent une liste de pays et leur score respectif.

2. Dans la liste **Case Sensitive**, sélectionnez **Yes** afin de prendre la casse en compte.

3. Sélectionnez le composant **tFileInputDelimited** et affichez son onglet **Basic settings.**

4. Pour renseigner le champ **File Name/Steam**, appuyez sur **Ctrl+Espace** pour accéder à la liste des variables. Pour traiter tous les fichiers du répertoire défini dans le composant **tFileList**, sélectionnez `tFileList.CURRENT_FILEPATH` dans la liste des variables globales.

5. Cliquez sur le bouton **Edit Schema** et définissez manuellement le schéma à deux colonnes pour qu'il corresponde au contenu des fichiers d'entrée.

tFileInputDelimited_1

Column	Key	Type	Nullable	Date...	Len...	Pre...	De...	Comment
Country	☐	String	☑					
Points	☐	Integer	☑					

Pour cet exemple, les deux colonnes sont *Country* et *Points*. Elles peuvent toutes les deux prendre une valeur nulle, ainsi cochez les cases de la colonne **Nullable**. La colonne *Country* sera de type **String**, et la colonne *Points* sera de type **Integer**.

6. Cliquez sur **OK** pour valider les paramètres, puis acceptez la propagation du schéma dans le reste du Job.

7. Puis sélectionnez le composant **tUnite** et affichez la vue **Component**. Notez que le schéma de sortie reflète exactement celui d'entrée et qu'il est en lecture seule.

8. Dans la vue **Component** du composant **tLogRow**, sélectionnez l'option **Table (Print values in cells of a table)** pour afficher correctement les valeurs de sortie.

Sauvegarder et exécuter le Job

1. Enregistrez le Job en appuyant sur les touches **Ctrl+S**.

2. Appuyez sur la touche **F6** ou cliquez sur le bouton **Run**, dans la vue **Run**, afin d'exécuter le Job.

   La console affiche les données des différents fichiers, fusionnées dans une seule table.

```
Starting job fetch at 16:11 08/03/2010.

.-------+------.
|  tLogRow_1   |
|=------+-----=|
|Country|Points|
|=------+-----=|
|france |12    |
|usa    |10    |
|france |14    |
|uk     |13    |
|usa    |9     |
|france |5     |
|france |16    |
|uk     |13    |
'-------+------'
Job fetch ended at 16:11 08/03/2010. [exit code=0]
```

tSleep

Propriétés du tSleep

Famille de composant	Orchestration	
Fonction	Le composant **tSleep** effectue une pause lors de l'exécution d'un Job.	
Objectif	Ce composant permet d'identifier les goulots d'étranglement éventuels à l'aide d'une pause lors de l'exécution du Job à des fins de test ou de monitoring. En production, il peut être utilisé pour tout type de pause dans le traitement d'un Job, pour alimenter un flux entrant par exemple.	
Basic settings	*Pause (in second)*	Temps en secondes pendant lequel le Job est suspendu.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le composant **tSleep** est généralement utilisé comme composant intermédiaire mettant un Job en pause, avant de le reprendre.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un exemple d'utilisation du **tSleep**, consultez *Scénario : Exécuter un Job à partir d'une boucle*.

tWaitForFile

Propriétés du tWaitForFile

Famille de composant	Orchestration	
Fonction	Le composant **tWaitForFile** fait une boucle sur un répertoire donné et déclenche le composant suivant lorsque la condition définie est rencontrée.	
Objectif	Ce composant est utilisé pour mettre en attente le composant lié. Il déclenche ce composant lorsque l'opération sur le fichier défini est effectuée dans le répertoire spécifié.	
Basic settings	*Time (in seconds) between iterations*	Paramétrez l'intervalle de temps entre deux itérations (en secondes).
	Max. number of iterations (infinite loop if empty)	Configurez le nombre maximal d'itérations à effectuer sur le répertoire spécifié.
	Directory to scan	Spécifiez le répertoire sur lequel effectuer des itérations.
	File mask	Saisissez le masque du fichier recherché afin de filtrer les fichiers à monitorer. Pour monitorer un fichier, saisissez son nom.
	Include subdirectories	Cochez cette case pour inclure les sous-répertoires dans l'itération.
	Case sensitive	Cochez cette case pour tenir compte de la casse.
	Include present file	Cochez cette case pour inclure le fichier courant.
	Trigger action when	Sélectionnez l'une des conditions de déclenchement : **a file is created** : déclenche le composant suivant lorsqu'un fichier est créé. **a file is deleted** : déclenche le composant suivant lorsqu'un fichier est supprimé. **a file is updated** : déclenche le composant suivant lorsqu'un fichier est modifié. **a file is created or updated or deleted** : déclenche le composant suivant lorsqu'un fichier est créé, supprimé ou modifié.
	Then	Sélectionnez l'action à effectuer avec la boucle itérative après déclenchement du composant suivant : - **continue loop** : continuer l'itération jusqu'à ce que le nombre maximum d'itérations soit atteint. - **exit loop** : arrêter l'itération immédiatement.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est local (built-in) et ne peut être modifié.
Advanced settings	*Wait for file to be released Every (in ms)*	Cochez cette case afin que le composant suivant ne se déclenche qu'après la fin de l'opération.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **CURRENT_ITERATION** : numéro de séquence de l'itération courante. Cette variable est une variable **Flow** et retourne un entier. **PRESENT_FILE** : nom et chemin d'accès au fichier. Cette variable est une variable **Flow** et retourne une chaîne de caractères.	

	DELETED_FILE : nom et chemin d'accès au fichier supprimé. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **CREATED_FILE** : nom et chemin d'accès au fichier créé. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **UPDATED_FILE** : nom et chemin d'accès au fichier mis à jour. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **FILENAME** : nom du fichier traité. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **NOT_UPDATED_FILE** : nom et chemin d'accès au fichier n'ayant pas été mis à jour. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de déclenchement du composant suivant, en fonction de la condition établie. Ainsi, ce composant doit être connecté à un autre composant.	
Connections		Liens de sortie (de ce composant à un autre) : **Row** : Main, Iterate. **Trigger** : OnSubjobOk, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate. **Trigger** : OnSubjobOk, Run if, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Attendre la création d'un fichier et arrêter l'itération après le déclenchement d'un message

Ce scénario décrit un Job scannant un répertoire et déclenchant un message affichant dans la console les informations de création d'un fichier lorsqu'un fichier texte est créé dans ce répertoire. Le scan prend fin et un autre message affiche le nombre d'itérations terminées.

Construire le Job

1. Glissez-déposez trois composants de la **Palette** dans l'espace de modélisation graphique : un **tWaitForFile** et deux composants **tJava**.
2. Reliez le **tWaitForFile** au premier **tJava** à l'aide d'un lien **Row** > **Iterate**.
3. Reliez le **tWaitForFile** au second **tJava** à l'aide d'un lien **Trigger** > **On Subjob Ok**.

Configurer les composants

1. Double-cliquez sur le **tWaitForFile** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **Directory to scan**, saisissez le nom du répertoire à scanner.
3. Dans le champ **File mask**, saisissez *"*.txt"* pour scanner uniquement les fichiers texte.
4. Dans la liste déroulante **Trigger action when**, sélectionnez **a file is created** afin de déclencher le premier **tJava** lorsqu'un fichier texte est créé.
5. Dans la liste déroulante **Then**, sélectionnez **exit loop** afin d'arrêter la boucle itérative dès que le premier **tJava** a été déclenché.

6. Double-cliquez sur le premier **tJava** pour ouvrir sa vue **Basic settings**.

7. Dans le champ **Code**, saisissez le code suivant :

```
System.out.println("A file was created at " + TalendDate.getCurrentDate());
System.out.println("Name of the created file: " +
 ((String)globalMap.get("tWaitForFile_1_CREATED_FILE")));
```

8. Double-cliquez sur le second **tJava** pour ouvrir sa vue **Basic settings**.

9. Dans le champ **code**, saisissez le code suivant :

```
System.out.println("\r\nIteration loop ended at " + TalendDate.getCurrentDate());
System.out.println("Number of iterations finished: " +
 ((Integer)globalMap.get("tWaitForFile_1_CURRENT_ITERATION")));
```

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter le Job.

3. Créez un fichier texte *exit.txt* dans le répertoire spécifié.

Lorsqu'un fichier texte est créé, un message concernant la création du fichier apparaît dans la console de la vue **Run**. L'itération se termine immédiatement et est suivie d'un autre message affichant l'heure de fin de la boucle itérative et le nombre d'itérations effectuées.

Scénario 2 : Attendre la création d'un fichier et poursuivre l'itération après le déclenchement d'un message

Basé sur le scénario précédent, ce scénario décrit un Job déclenchant un message dans la console ainsi que la création d'un fichier texte dans le répertoire scanné, sans arrêter le processus de scan immédiatement. Le Job continue à scanner le répertoire jusqu'à ce que le nombre d'itérations atteigne la valeur maximale définie.

1. Double-cliquez sur le **tWaitForFile** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **Time (in seconds) between iterations**, saisissez l'intervalle de temps en secondes entre deux itérations. Dans cet exemple, saisissez *5*.

3. Dans le champ **Max. number of iterations (infinite loop if empty)**, saisissez le nombre maximal d'itérations. Dans cet exemple, saisissez *10*.

4. Dans la liste **Then**, sélectionnez **continue loop** afin de continuer la boucle itérative, même après déclenchement du message.

5. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job.

6. Appuyez sur **F6** pour exécuter le Job.

7. Créez un fichier texte *continue.txt* dans le répertoire spécifié.

```
Starting job DOCT3129_tWaitForFile_S01 at 14:35 18/03/2014.

[statistics] connecting to socket on port 3455
[statistics] connected
A file was created at Tue Mar 18 14:35:28 CST 2014
Name of the created file: E:\Studio\continue.txt

The iteration was ended at Tue Mar 18 14:36:03 CST 2014
Number of iterations finished: 10
[statistics] disconnected
Job DOCT3129_tWaitForFile_S01 ended at 14:36 18/03/2014. [exit code=0]
```

Lorsqu'un fichier texte est créé, un message concernant la création du fichier est affiché dans la console de la vue **Run**. La boucle itérative est arrêtée jusqu'à ce que le nombre d'itérations atteigne dix. Un nouveau message s'affiche et indique l'heure de fin de la boucle itérative et le nombre d'itérations effectuées.

tWaitForSocket

Propriétés du tWaitForSocket

Famille de composant	Orchestration	
Fonction	Le composant **tWaitForSocket** fait une boucle sur un port défini à la recherche de données puis déclenche un sous-job lorsque la condition est rencontrée.	
Objectif	Ce composant permet de déclencher un Job en fonction d'une condition définie.	
Basic settings	*Port*	Saisissez le numéro du port à écouter.
	End of line separator	Saisissez le séparateur de fin de ligne de vos données.
	Then	Sélectionnez l'action à effectuer : soit continuer à écouter le port (**keep on listening**) soit fermer la connexion (**close socket**).
	Print client/server data	Cochez cette case afin d'afficher les données client ou serveur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant est un composant de début (ou de déclenchement) pour le sous-Job exécuté en fonction de la condition établie. Ainsi, ce composant doit être connecté à un sous-job via un lien Iterate.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **INPUT_DATA** : données transmises par le client. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.	

Scénario associé

Aucun scénario n'est disponible pour ce composant.

tWaitForSqlData

Propriété du tWaitForSqlData

Famille de composant	Orchestration	
Fonction	Le composant **tWaitForSqlData** effectue une boucle sur une connexion donnée à la recherche d'un ajout ou d'une suppression de lignes, puis déclenche un sous-Job lorsque la condition est rencontrée.	
Objectif	Ce composant permet de déclencher un sous-Job en fonction de la présence de données SQL.	
Basic settings	*Wait at each iteration (in seconds)*	Paramétrez en secondes l'intervalle de temps entre chaque vérification.
	Max. iterations (infinite if empty)	Nombre de vérifications à effectuer avant que le Job ne s'arrête (si vous ne saisissez rien, la boucle sera infinie).
	Use an existing connection/ Component List	Cochez la case et cliquez sur le composant de connexion adéquat dans la liste **Component list** pour réutiliser les informations de connexion que vous avez déjà définies. Lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component List** présente uniquement les composants de connexion du Job du même niveau. Si vous souhaitez utiliser une connexion existant dans un autre niveau, vérifiez que les composants de connexion disponibles partagent la même connexion. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez **Use or register a shared DB connection** dans tous les composants de base de données que vous utilisez. Sinon, vous pouvez également désactiver le composant de connexion et utiliser les **Dynamic settings** du composant, afin de paramétrer manuellement la connexion. Dans ce cas, vérifiez que le nom de la connexion est unique et distinct tout au long des deux niveaux de Jobs. Pour plus d'informations concernant les paramètres dynamiques (**Dynamic settings**), consultez le *Guide utilisateur du Studio Talend*.
	Table to scan	Nom de la table à vérifier.
	Trigger action when rowcount is	Sélectionnez la condition à rencontrer pour exécuter l'action : **Equal to** : égal à **Not Equal to** : différent de **Greater than** : supérieur à **Lower than** : inférieur à **Greater or equal to** : supérieur ou égal à **Lower or equal to** :inférieur ou égal à.
	Value	Définissez la valeur à prendre en compte.
	Then	Sélectionnez l'action à effectuer lorsque la condition est rencontrée : soit arrêter la boucle (**exit loop**), soit continuer la boucle jusqu'à ce que le nombre d'itération maximum soit atteint (**continue loop**).

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **CURRENT_ITERATION** : numéro de séquence de l'itération courante. Cette variable est une variable **Flow** et retourne un entier. **ROW_COUNT** : nombre de lignes détectées dans la table. Cette variable est une variable **Flow** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Bien que ce composant ait besoin d'un composant de connexion pour ouvrir l'accès à une base de données, il joue aussi le rôle de composant de début (ou de déclenchement) pour le sous-Job à exécuter lorsque la condition est rencontrée. Ainsi, ce composant doit être connecté à un sous-Job via un lien Iterate.

Scénario : Attendre l'ajout de lignes dans une table

Ce scénario décrit un Job lisant la table d'une base de données dans l'attente d'un ajout de données dans cette table, pour ensuite exécuter un sous-Job. Lorsque des données sont ajoutées, le sous-Job effectue un Select* sur la table et affiche le contenu des données insérées dans la console standard.

- Cliquez et déposez les composants suivants de la **Palette** dans l'espace de modélisation : **tMysqlConnection**, **tWaitForSqlData**, **tMysqlInput**, **tLogRow**.
- Connectez le composant **tMysqlConnection** au composant **tWaitforSqlData** à l'aide un lien **OnSubjobOK**, disponible via le clic-droit.
- Puis connectez le composant **tWaitForSqlData** au sous-Job à l'aide d'un lien **Iterate** puisqu'aucune donnée n'est transférée pour le moment. En effet, le **tWaitForSqlData** effectue seulement des boucles jusqu'à ce que la condition soit rencontrée.
- Dans le sous-Job à exécuter si la condition est rencontrée, le composant **tMysqlInput** est connecté au **tLogRow**. Puisque cette connexion sert à transférer des données, utilisez un lien de type **Row main**.
- Maintenant, paramétrez la connexion à la table devant être vérifiée à intervalles réguliers. Dans l'onglet **Basic settings** de la vue **Component** du composant **tMysqlConnection**, paramétrez les propriétés de connexion à la base de données.

- Renseignez les champs **Host**, **Port**, **Database**, **Username**, **Password** pour ouvrir une connexion à la base de données. Pour plus d'informations, consultez les composants de connexion aux bases de données. Par exemple, pour une base de données Mysql, consultez *tMysqlConnection*.
- Dans le champ **Encoding**, sélectionnez l'encodage adéquat, si nécessaire.
- Puis sélectionnez le composant **tWaitForSqlData**, et dans l'onglet **Basic settings** de la vue **Component**, paramétrez ses propriétés.
- Dans le champ **Wait at each iteration**, paramétrez en secondes l'intervalle entre chaque itération.

- Dans le champ **Max iterations**, saisissez le nombre d'itérations maximum à effectuer avant que le Job ne s'arrête.
- Pour utiliser le composant **tWaitForSqlData**, il faut qu'une connexion soit ouverte pour effectuer le nombre de boucles défini. Sélectionnez la connexion adéquate (s'il y en a plusieurs) dans la liste déroulante **Component List**.
- Dans le champ **Table to scan**, saisissez le nom de la table à scanner. Dans cet exemple : *test_datatypes*.
- Dans les champs **Trigger action when rowcount is** et **Value**, sélectionnez la condition à rencontrer afin de lancer le sous-Job. Pour cet exemple, le nombre de lignes à prendre en compte dans la table scannée soit être supérieur ou égal à 1, ainsi sélectionnez *be greater or equal to 1*.
- Dans le champ **Then**, sélectionnez l'action à effectuer lorsque la condition est rencontrée avant que le nombre maximum d'itérations défini ne soit atteint. Dans cet exemple, dès que la condition est rencontrée, la boucle devrait s'arrêter.
- Puis paramétrez le sous-Job à exécuter lorsque la condition est rencontrée. Dans cet exemple, le sous-Job sélectionne les données de la table scannée et les affiche dans le console.
- Sélectionnez le composant **tMysqlInput**, et dans l'onglet **Basic settings** de la vue **Component**, paramétrez la connexion à la table.

- Si la connexion est stockée dans le **Repository**, sélectionnez-la dans la liste déroulante. Sinon, cochez la case **Use an existing connection** et sélectionnez le composant de connexion correspondant dans la liste.
- Dans cet exemple, le schéma correspondant à la structure de la table est stocké dans le **Repository**.
- Dans le champ **Table Name**, renseignez le nom de la table à partir de laquelle les données seront extraites, *Test_datatypes*.
- Puis dans le champ **Query**, saisissez la commande `Select` permettant d'extraire le contenu de la table.
- Pour cet exemple, laissez les propriétés par défaut du composant **tLogRow**.

Avant d'exécuter le Job, assurez-vous que la table à scanner (*test_datatypes*) est bien vide, afin que la condition (*greater or equal to 1*) soit rencontrée. Puis exécutez le Job via le raccourci **F6**. Avant la fin de la boucle, ajoutez une ou deux lignes à la table *test_datatypes* afin que la condition soit rencontrée.

id	log_msg
2	143.112.32.4 - - [04/Mar/2008:00:00:00 +0100] "GET /c...
1	143.112.32.4 - - [04/Mar/2008:00:00:00 +0100] "GET /c...

Le Job s'arrête lorsque les données ajoutées à la table sont détectées lors de la boucle et le contenu de la table est affiché dans la console.

```
Starting job tWaitForSql at 16:55 06/03/2008.
2| 143.112.32.4 - - [04/Mar/2008:00:00:00 +0100] "GET
/components/none HTTP/1.1" 404 354 "-" "Mozilla/4.0 (compatible
1| 143.112.32.4 - - [04/Mar/2008:00:00:00 +0100] "GET
/components/none HTTP/1.1" 404 354 "-" "Mozilla/4.0 (compatible
Job tWaitForSql ended at 16:55 06/03/2008. [exit code=0]
```

Composants Processing

Ce chapitre passe en revue les propriétés des principaux composants présents dans la famille **Processing** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **Processing** regroupe les composants vous permettant d'effectuer différents types de traitement de flux de données, notamment l'agrégation, le mapping, la transformation, la dénormalisation, le filtrage, etc.

tAggregateRow

Propriétés du tAggregateRow

Famille de composant	Processing	
Fonction	Le composant **tAggregateRow** reçoit un flux de données et fait une agrégation basée sur une ou plusieurs colonnes. Pour chacune des lignes en sortie, une clé d'agrégation est fournie, ainsi que le résultat de l'opération d'agrégation correspondant (min, max, sum, etc.).	
Objectif	Il permet d'établir des métriques et des statistiques basées sur des valeurs ou des calculs.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Group by	Définit les ensembles d'agrégation, dont les valeurs sont utilisées pour les calculs.
		Output Column : Sélectionnez le libellé de colonne dans la liste fournie, basée sur la structure de schéma que vous avez définie. Vous pouvez ajouter autant de colonnes de sortie que vous le souhaitez afin d'affiner les agrégations. Ex : Sélectionnez *Country* (Pays) pour calculer la moyenne des valeurs pour chaque pays ou sélectionnez *Country* et *Region* pour comparer les résultats des régions d'un pays par rapport aux régions d'un autre pays.
		Input Column : Faites la correspondance entre les libellés des colonnes d'entrée avec ceux des colonnes de sortie, dans le cas où vous souhaitez que les libellés du schéma de sortie soient différents du schéma d'entrée.
	Operations	Sélectionnez le type d'opération à effectuer ainsi que la valeur à utiliser pour le calcul et le champ de sortie.
		Output Column : Sélectionnez le champ de destination dans la liste.
		Fonction : Sélectionnez l'opérateur parmi :

		count : calcule le nombre de ligne, **min :** sélectionne la plus petite valeur, **max :** sélectionne la plus grande valeur, **avg :** calcule la moyenne, **sum :** calcule la somme, **first :** retourne la première valeur, **last :** retourne la dernière valeur, **list :** liste les valeurs d'une agrégation de plusieurs clés. **list (object) :** liste les valeurs Java d'une agrégation de plusieurs clés, **count (distinct) :** compte le nombre de lignes sans les doublons, **standard deviation :** calcule l'écart-type.
		Input column position : Sélectionnez la colonne d'entrée à partir de laquelle les valeurs sont collectées pour l'agrégation.
		Ignore null values : Cochez cette case devant le nom de toutes les colonnes pour lesquelles vous voulez que les valeurs nulles soient ignorées.
Advanced settings	*Delimiter(only for list operation)*	Saisissez le séparateur que vous souhaitez utiliser afin de séparer les différentes opérations.
	Use financial precision, this is the max precision for "sum" and "avg" operations, checked option heaps more memory and slower than unchecked.	Cochez cette case afin d'utiliser une précision financière, une précision maximale, mais qui consomme plus de mémoire et rend le processus plus lent. *Il est conseillé d'utiliser le type BigDecimal en sortie avec cette option, afin d'obtenir des résultats précis.*
	Check type overflow (slower)	Vérifie les types des données afin d'éviter que le Job ne plante
	Check ULP (Unit in the Last Place), ensure that a value will be incremented or decremented correctly, only float and double types. (slower)	Cochez cette case afin d'avoir la meilleure précision possible pour les types Float et Double.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n'est pas disponible dans la version Map/Reduce de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire car il traite un flux de données, par conséquent il requiert des composants en entrée comme en sortie. Généralement, l'utilisation du composant **tAggregateRow** est combinée au composant **tSortRow**.	
Utilisation dans les Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop.	

	Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Utilisation dans des Jobs Storm	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez également utiliser ce composant en tant que composant Storm. Dans un Job Storm **Talend**, ce composant est utilisé comme étape intermédiaire. Les autres composants utilisés avec celui-ci doivent également être des composants Storm. Ils génèrent nativement du code Storm pouvant être exécuté directement dans un système Storm. La version de Storm ne supporte pas l'utilisation des variables globales. Vous devez utiliser l'onglet **Storm Configuration** dans la vue **Run** pour définir la connexion à un système Storm donné pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant un Job Storm **Talend**, consultez les sections décrivant comment créer et configurer un Job Storm **Talend** dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf lorsqu'explicitement indiqué, un scénario présente uniquement des Jobs de type **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Agréger des valeurs et trier des données

Le scénario suivant décrit un Job à quatre composants. Le composant d'entrée, un fichier délimité CSV, contient des noms de pays et des valeurs de notation à trier par ordre décroissant de moyenne. Ce composant d'entrée est connecté à un opérateur **tAggregateRow** qui se charge de calculer les moyennes puis à un composant **tSortRow** qui se charge du tri. Le flux de sortie est dirigé dans un nouveau fichier CSV.

- A partir du dossier **File** de la **Palette**, cliquez et déposez un composant **tFileInputDelimited**.
- Cliquez sur le libellé du composant et renommez-le en *Countries*. Ou passez dans l'onglet **View** pour le changer.
- Dans l'onglet **Basic settings**, définissez le chemin d'accès ainsi que les critères de délimitation. Ou sélectionnez l'entrée de métadonnées correspondant à votre fichier d'entrée, si vous en avez défini une.
- Cliquez sur **Edit schema...** et paramétrez les colonnes : *Countries* et *Points* correspondant à la structure du fichier. Si la description du fichier est conservée dans les **Metadata** du **Repository**, le schéma est automatiquement chargé lorsque vous cliquez sur **Repository** dans le champ **Schema**.

- Puis dans le dossier **Processing** de la **Palette**, cliquez et déposez le composant **tAggregateRow**. Renommez-le en *Calculation*.
- Cliquez-droit et connectez *Countries* à *Calculation* à l'aide d'une connexion **Row > Main**.
- Double-cliquez sur *Calculation* (composant **tAggregateRow**) pour définir les propriétés. Cliquez sur **Edit schema** et définissez le schéma de sortie. Vous pouvez ajouter autant de colonnes que vous le voulez selon le nombre d'opérations d'ensemble que vous souhaitez obtenir en sortie.

- Dans cet exemple, calculez la moyenne de la notation par pays et affichez les valeurs minimale et maximale par pays, sachant que chaque pays possède plusieurs notes. Cliquez sur **OK** lorsque le schéma est terminé.
- Définissez à présent les différentes opérations d'ensemble à effectuer. Dans le champ **Group By** de la fenêtre **[Basic settings]** du composant **tAggregateRow**, définissez les ensembles sur lesquels les opérations sont effectuées. Dans cet exemple, sélectionnez **Country** comme colonne de regroupement. Notez que dans la colonne de sortie, un champ clé doit être défini dans le schéma. La première colonne citée comme de sortie (Output) dans la table **Group By** devient l'ensemble principal de calcul. Toutes les autres sorties deviennent alors secondaires dans l'ordre d'affichage.
- Sélectionnez la colonne d'entrée (Input) dont sont extraites les valeurs.
- Puis renseignez les diverses opérations réalisées. Les fonctions utilisées dans cet exemple sont des moyennes (average), min, max. Sélectionnez la colonne d'entrée dont sont extraites les valeurs de calcul et sélectionnez les cases **Ignore null values** si nécessaire.

- Cliquez-déposez un composant **tSortRow** dans l'espace de modélisation. Pour plus d'informations concernant ce composant, consultez *Propriétés du tSortRow*
- Connectez-le au composant **tAggregateRow** à l'aide d'une connexion de type **Row > Main**.
- Dans la vue **Component** du composant **tSortRow**, définissez la colonne de tri, le type ainsi que l'ordre de tri.

- Dans ce scénario, la colonne à trier est *Country*, le type de tri est alphabétique et l'ordre est ascendant.
- Ajoutez un composant de sortie à ce Job. Cliquez et déposez un composant **tFileOutputDelimited** et paramétrez-le.
- Connectez le composant **tSortRow** à ce composant de sortie.
- Dans la vue **Component**, saisissez le chemin d'accès au fichier de sortie. Cliquez sur **Edit schema** pour le configurer si nécessaire. Dans ce scénario, le fichier délimité est de type CSV.

  Cochez la case **Include Header** afin de réutiliser les libellés des colonnes du schéma dans le flux de sortie.
- Appuyez sur **F6** pour exécuter le Job. Le fichier CSV est ainsi créé et il contient les résultats des opérations d'agrégation.

tAggregateSortedRow

Propriétés du tAggregateSortedRow

Famille du composant	Processing	
Fonction	Le composant **tAggregateSortedRow** reçoit un flux de données triées sur lequel il effectue une agrégation basée sur une ou plusieurs colonnes. La clé d'agrégation et les résultats pertinents des opérations (min, max, sum, etc.) sont dispensés pour chaque ligne de sortie.	
Objectif	Le **tAggregateSortedRow** vous permet de récupérer un échantillon de métriques reposant sur des valeurs ou des calculs. Etant donné que les données du flux d'entrée sont supposées être déjà triées, les performances n'en sont que plus optimisées.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent connecté dans le Job.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job Designs. Voir également le *Guide utilisateur du Studio Talend*.
	Input rows count	Spécifiez le nombre de lignes envoyé au composant **tAggregateSortedRow**. Si vous spécifiez une limite (dans le champ **Limit**) au nombre de lignes à traiter dans le composant d'entrée, vous devez utiliser cette même limite dans le champ **Input rows count**.
	Group by	Définit les ensembles d'agrégation, dont les valeurs sont utilisées pour les calculs.
		Output Column : Sélectionnez le libellé de colonne dans la liste fournie, basée sur la structure de schéma que vous avez définie. Vous pouvez ajouter autant de colonnes de sortie que vous le souhaitez afin d'affiner les agrégations. Exemple : Sélectionnez *Country* (Pays) pour calculer la moyenne des valeurs pour chaque pays ou sélectionnez *Country* et *Region*

		pour comparer les résultats des régions d'un pays par rapport aux régions d'un autre pays.
		Input Column : Faites la correspondance entre les libellés des colonnes d'entrée avec ceux des colonnes de sortie, dans le cas où vous souhaitez que les libellés du schéma de sortie soient différents du schéma d'entrée.
	Operations	Sélectionnez le type d'opération à effectuer ainsi que la valeur à utiliser pour le calcul et le champ de sortie.
		Output Column : Sélectionnez le champ de destination à partir de la liste déroulante.
		Function : Choisissez l'opérateur parmi : **count**, **min**, **max**, **avg**, **first**, **last**.
		Input column position : Sélectionnez la colonne d'entrée à partir de laquelle les valeurs sont récupérées avant d'être agrégées.
		Ignore null values : Cochez cette case devant le nom de toutes les colonnes pour lesquelles vous voulez que les valeurs nulles soient ignorées.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant traite des flux de données, il nécessite donc des composants d'entrée et sortie. Il est donc défini en tant que composant intermédiaire.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario associé

Pour un scénario associé, consultez *Scénario : Agréger des valeurs et trier des données* du composant **tAggregateRow**.

tConvertType

Propriétés du tConvertType

Famille de composant	Processing	
Fonction	Le composant **tConvertType** permet d'effectuer des conversions spécifiques du type de données Java vers un autre type de données Talend au cours de l'exécution.	
Objectif	Le composant **tConvertType** permet de convertir automatiquement des données de type Java vers un autre type utilisé par **Talend**, ce qui permet d'éviter les erreurs de compilation.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé dans divers projets et Jobs designs. Voir également le *Guide utilisateur du Studio Talend*.
	Auto Cast	Cette case est cochée par défaut. Elle permet de convertir automatiquement des données de type Java.
	Manual Cast	Ce mode n'est pas visible si la case **Auto Cast** est cochée. Elle permet de préciser manuellement les colonnes où une conversion de type Java est nécessaire.
	Set empty values to Null before converting	Cochez cette case pour définir les valeurs des éléments de type **String** ou **Object** comme null pour les données d'entrée.
	Die on error Indisponible pour les Jobs Map/Reduce.	Cette case est cochée pour arrêter le Job lorsque survient une erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n'est pas disponible dans la version Map/Reduce de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant ne peut être utilisé comme composant de début. Il requiert un flux d'entrée pour fonctionner.
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.

Scénario 1 : Convertir des types de données Java

Le scénario Java suivant permet de créer un Job à quatre composants dans lequel le **tConvertType** est utilisé pour convertir des données de type Java stockées dans trois colonnes différentes et le **tMap** est utilisé pour modifier le schéma : la première colonne reste intacte et les deux dernières colonnes du schéma d'entrée sont fusionnées en une seule colonne dans le schéma de sortie.

Dans ce scénario, le schéma du fichier délimité utilisé en entrée est stocké dans le Repository, vous pouvez donc renseigner les propriétés du composant **tFileInputDelimited** d'un simple glisser-déposer du répertoire **Repository** > **Metadata** > **File delimited** vers l'espace de modélisation. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.

Déposer les composants

1. Cliquez et déposez les composants suivants dans l'espace de modélisation : **tConvertType**, **tMap** et **tLogRow**.

2. Dans le Repository, développez les nœuds **Metadata > File Delimited** et glissez-déposez la métadonnée correspondante dans l'espace de modélisation. Pour ce scénario, *JavaTypes*.

   La boîte de dialogue **[Components]** apparaît.

3. Dans la boîte de dialogue, sélectionnez le **tFileInputDelimited** dans la liste des composants et cliquez sur **Ok**.

   Un composant **tFileInputComponent** appelé *Java types* apparaît dans l'espace de modélisation.

4. Connectez les composants à l'aide de liens **Row Main**.

Configurer les composants

1. Dans l'espace de modélisation, sélectionnez le composant **tFileInputDelimited** et cliquez sur la vue **Component** pour paramétrer ses propriétés de base.

2. Dans l'onglet **Basic settings**, sélectionnez l'option **Repository** dans la liste **Property Type** puisque les propriétés du fichier sont stockées dans le **Repository**. Les autres champs sont alors renseignés automatiquement grâce aux données récupérées.

Le fichier d'entrée utilisé pour ce scénario est un fichier texte appelé *input* contenant des données Java de type string (chaîne de caractères), integer (entier) et float.

Renseignez les autres champs, si nécessaire. Pour plus d'informations, consultez *tFileInputDelimited*. Dans ce scénario, il n'y a ni en-tête ni pied de page et il n'y a pas de nombre limite de lignes à traiter.

3. Cliquez sur **Edit schema** pour décrire la structure des données du fichier d'entrée. Dans ce scénario, le schéma est composé de trois colonnes : *StringtoInteger*, *IntegerField* et *FloatToInteger*.

4. Cliquez sur **OK** pour fermer la boîte de dialogue.

5. Dans l'espace de modélisation, sélectionnez le **tConvertType** et cliquez sur la vue **Component** pour paramétrer ses propriétés de base.

6. Dans la liste **Schema Type**, sélectionnez l'option **Built in** et cliquez sur le bouton **Sync columns** pour récupérer automatiquement les colonnes du composant **tFileInputDelimited**.

7. Si nécessaire, cliquez sur **Edit schema** pour décrire manuellement la structure des données de ce composant de traitement.

Dans ce scénario, l'objectif est de convertir les données de type string (chaîne de caractères) en integer (entier) et des données de type float en integer.

Cliquez sur **OK** pour fermer la boîte de dialogue **[Schema of tConvertType]**.

8. Dans l'espace de modélisation, double-cliquez sur le **tMap** pour ouvrir l'éditeur du **tMap**.

L'éditeur du **tMap** s'ouvre et affiche les noms des colonnes du fichier d'entrée.

9. Dans l'onglet **Schema editor** en bas de l'éditeur du **tMap**, cliquez sur le bouton **[+]** de la table de sortie pour ajouter deux lignes et nommez-les *StringToInteger* et *Sum*.

10. Dans l'éditeur du **tMap**, glissez la ligne *StringToInteger* de la table d'entrée vers la ligne *StringToInteger de la table de sortie*.

11. Dans l'éditeur du **tMap**, glissez les lignes *IntegerField* et *FloatToInteger* de la table d'entrée vers la ligne *Sum* de la table de sortie. Cliquez sur **OK** pour fermer l'éditeur du **tMap**.

12. Dans l'espace de modélisation, sélectionnez le **tLogRow** et cliquez sur la vue **Component** pour paramétrer ses propriétés de base. Pour plus d'informations, consultez *tLogRow*.

Exécuter le Job

1. Appuyez sur **Ctrl+S** pour enregistrer votre Job.
2. Appuyez sur **F6** pour l'exécuter.

```
Starting job tConvertType at 18:03 19/11/2008.
.---------------+---.
|     tLogRow_1     |
|=---------------+--=|
|StringToInteger|sum|
|=---------------+--=|
|3              |579|
'---------------+---'

Job tConvertType ended at 18:03 19/11/2008. [ex
```

Les données de type string sont converties en integer et affichées dans la colonne *StringToInteger* dans la console. Les données de type float sont converties en integer et ajoutées à la valeur *IntegerField* et leur somme est affichée dans la colonne *Sum* dans la console.

Scénario 2 : Convertir des types Java à l'aide de composants Map/Reduce

Si vous êtes utilisateur d'une solution Big Data en souscription, vous pouvez produire la version Map/Reduce du Job décrit précédemment, à l'aide de composants Map/Reduce. Ce Job Map/Reduce **Talend** génère du code Map/Reduce et s'exécute nativement dans Hadoop.

Les données d'exemple utilisées dans ce scénario sont les mêmes que dans le scénario précédent.

```
3;123;456.21
```

Comme le *Studio Talend* vous permet de convertir un Job d'une version **Map/Reduce** à une version **Standard** (Non Map/Reduce) et vice-versa, vous pouvez convertir le scénario précédent afin de créer ce Job Map/Reduce. Ainsi, de nombreux composants utilisés conservent leurs paramètres originaux afin de réduire votre charge de travail lors de la conception du Job.

Avant de commencer à reproduire ce scénario, assurez-vous d'avoir les droits et permissions appropriés pour accéder à la distribution Hadoop à utiliser. Procédez comme suit :

Convertir le Job

1. Dans le **Repository** de la perspective **Integration** du *Studio Talend*, cliquez-droit sur le Job que vous avez créé dans le scénario précédent afin d'ouvrir son menu contextuel. Sélectionnez ensuite **Edit properties**.

L'assistant **[Edit properties]** s'ouvre alors. Notez que le Job doit être fermé avant que vous ne puissiez effectuer des changements dans cet assistant.

L'assistant se présente comme dans la capture d'écran ci-dessous :

Notez que vous pouvez changer le nom du Job ainsi que les autres informations de description du Job à partir de cet assistant.

2. Cliquez sur **Convert to Map/Reduce Job**. Un Job Map/Reduce portant le même nom apparaît sous le sous-nœud **Map/Reduce Jobs** du nœud **Job Design**.

Si vous devez partir d'un Job Map/Reduce Job vierge, cliquez-droit sur le nœud **Job Design** ou sur le sous-nœud **Map/Reduce Jobs** et sélectionnez **Create Map/Reduce Job** dans le menu contextuel. Un Job vierge s'ouvre dans l'espace de modélisation graphique. Pour plus d'informations, consultez la section décrivant comment créer un Job Map/Reduce dans le *Guide de prise en main de Talend Open Studio for Big Data*.

Reconstruire le Job

1. Double-cliquez sur ce nouveau Job Map/Reduce pour l'ouvrir dans l'espace de modélisation graphique. La **Palette** des composants Map/Reduce s'ouvre. Dans l'espace de modélisation graphique, si des composants sont barrés, c'est qu'ils n'ont pas de version Map/Reduce.

2. Cliquez-droit sur chaque composant barré et sélectionnez **Delete** afin de les supprimer de l'espace de modélisation graphique.

3. Déposez un composant **tHDFSInput** dans l'espace de modélisation. Le **tHDFSInput** lit des données de la distribution Hadoop utilisée.

   Si vous construisez le Job à partir de rien, déposez également un **tConvertType**, un **tMap** et un **tLogRow**.

4. Reliez le **tHDFSInput** au **tConvertType** à l'aide d'un lien **Row > Main** et acceptez la propagation du schéma du **tConvertType**.

Configurer une connexion Hadoop

1. Cliquez sur **Run** afin d'ouvrir la vue correspondante et cliquez sur l'onglet **Hadoop Configuration** afin d'afficher cette vue et configurer la connexion Hadoop pour ce Job.

   La vue se présente comme dans la capture d'écran ci-dessous :

2. Dans la liste **Property type**, sélectionnez **Built-in**. Si vous avez créé une connexion afin qu'elle soit utilisable depuis le **Repository**, sélectionnez alors **Repository** afin que le *Studio Talend* réutilise les informations de connexion pour ce Job.

   Pour plus d'informations concernant la création d'une connexion Hadoop dans le **Repository**, consultez le chapitre décrivant le nœud **Hadoop cluster** du *Guide de prise en main de Talend Open Studio for Big Data*.

3. Dans la zone **Version**, sélectionnez la distribution Hadoop à utiliser et sa version. Si vous ne trouvez pas la distribution correspondant à la votre dans la liste, sélectionnez **Custom** afin de vous connecter à une distribution Hadoop non officiellement supportée par le *Studio Talend*.

   Pour un exemple d'utilisation étape par étape de cette option **Custom**, consultez *Connexion à une distribution Hadoop personnalisée*.

En plus de l'évolution de Hadoop, veuillez noter les changements suivants :

- Si vous utilisez **Hortonworks Data Platform V2.2**, les fichiers de configuration de votre cluster peuvent utiliser des variables d'environnement comme *${hdp.version}*. Si vous êtes dans cette situation, vous devez configurer la propriété *mapreduce.application.framework.path* dans la table **Hadoop properties**, avec la valeur du chemin d'accès pointant explicitement vers l'archive du framework MapReduce de votre cluster. Par exemple :

```
mapreduce.application.framework.path=/hdp/apps/2.2.0.0-2041/mapreduce/
mapreduce.tar.gz#mr-framework
```

- Si vous utilisez **Hortonworks Data Platform V2.0.0**, le type de système d'exploitation utilisé pour la distribution Hadoop et pour le Job **Talend** doit être identique, par exemple Windows ou Linux. Sinon, vous devez utiliser le Jobserver de **Talend** afin d'exécuter un Job dans le même type de système d'exploitation que celui dans lequel la distribution **Hortonworks Data Platform V2.0.0** utilisée est exécutée. Pour plus d'informations concernant le Jobserver de **Talend**, consultez le *Guide d'installation et de migration Talend*.

4. Dans le champ **Name node**, saisissez l'emplacement du nœud maître, le NameNode, de la distribution utilisée. Par exemple, *hdfs://tal-qa113.talend.lan:8020*.

   Si vous utilisez une distribution MapR, vous pouvez simplement laisser *maprfs:///* dans ce champ. Le client MapR se charge de créer la connexion au moment de l'exécution. Notez que le client doit être correctement installé. Pour plus d'informations concernant la configuration d'un client MapR, consultez la page suivante sur le site de la documentation de MapR : http://doc.mapr.com/display/MapR/Setting+Up+the+Client (en anglais).

5. Dans le champ **Job tracker**, saisissez l'emplacement du JobTracker de votre distribution. Par exemple, *tal-qa114.talend.lan:8050*.

   Notez que le mot "Job" dans le terme "JobTracker" désigne les Jobs MR ou Map/Reduce décrits dans la documentation Apache disponible sur le site Internet d'Apache http://hadoop.apache.org/ (en anglais).

   Si vous utilisez YARN dans votre cluster Hadoop, comme **Hortonworks Data Platform V2.0.0** ou **Cloudera CDH4.3 + (mode YARN)**, vous devez spécifier l'emplacement du **Resource Manager** et non celui du Jobtracker. Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser :

   - Cochez la case **Set resourcemanager scheduler address** et saisissez l'adresse de l'ordonnanceur (Scheduler) dans le champ qui apparaît.

   - Cochez la case **Set jobhistory address** et saisissez l'emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory.

   - Cochez la case **Set staging directory** et saisissez le chemin d'accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l'exécution de programmes. Ce répertoire se trouve sous la propriété *yarn.app.mapreduce.am.staging-dir* dans les fichiers de configuration, notamment les fichiers *yarn-site.xml* et *mapred-site.xml* de votre distribution.

   - Cochez la case **Use datanode hostname** pour permettre au Job d'accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété *dfs.client.use.datanode.hostname* à *true*. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.

6. Si la distribution utilisée nécessite une authentification Kerberos, cochez la case **Use Kerberos authentication** et saisissez les détails d'authentification. Sinon, laissez cette case décochée.

   Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d'utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

De plus, comme ce composant effectue des calculs Map/Reduce, vous devez également authentifier les services associés, comme le serveur de l'historique des Jobs et le gestionnaire de ressources ou le JobTracker, selon votre distribution, dans le champ correspondant. Ces principaux se trouvent dans les fichiers de configuration de votre distribution. Par exemple, dans une distribution CDH4, le principal du gestionnaire de ressource est configuré dans le fichier *yarn-site.xml* et le principal de l'historique des Job dans le fichier *mapred-site.xml*.

Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case **Use a keytab to authenticate**. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**.

Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.

7. Dans le champ **User name**, saisissez votre nom d'utilisateur pour votre distribution. Si vous laissez ce champ vierge, le nom d'utilisateur de la machine hébergeant le *Studio Talend* est utilisé.

8. Dans le champ **Temp folder**, saisissez le chemin d'accès au dossier HDFS dans lequel sont stockés les fichiers temporaires générés pendant les opérations de Map/Reduce.

9. Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d'autres termes, changez le séparateur si celui-ci n'est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.

10. Laissez la case **Clear temporary folder** (Effacer le contenu du dossier temporaire) cochée, sauf si vous souhaitez garder ces fichiers temporaires.

11. Laissez la case **Compress intermediate map output to reduce network traffic** cochée, afin de réduire le temps de transfert des partitions du Mapper aux différents Reducers.

    Cependant, si le transfert des données dans le Job est négligeable, il est recommandé de décocher cette case afin de désactiver l'étape de compression, car la compression consomme des ressources supplémentaires du CPU.

12. Si vous souhaitez utiliser des propriétés Hadoop personnalisées, renseignez la table **Hadoop properties** avec la ou les propriété(s) à personnaliser. Lors de l'exécution, ces modifications écrasent les propriétés correspondantes par défaut utilisées par le Studio pour son moteur Hadoop.

    Pour plus d'informations concernant les propriétés requises par Hadoop, consultez la documentation Apache Hadoop sur le site http://hadoop.apache.org (en anglais), ou la documentation de la distribution Hadoop que vous utilisez.

13. Si le système Hadoop utilisé est Hortonworks Data Platform V1.2 ou Hortonworks Data Platform V1.3, vous devez définir des allocations de mémoire suffisantes pour que les opérations de map et reduce soient effectuées par le système Hadoop.

    Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations *map* et *reduce* dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux *1000*, ce qui est normalement adapté pour l'exécution de ces opérations.

    Si la distribution utilisée est YARN, les paramètres de mémoire à définir sont alors **Map (in Mb)**, **Reduce (in Mb)** et **ApplicationMaster (in Mb)**, respectivement. Ces champs permettent d'allouer dynamiquement de la mémoire aux opérations map et reduce et à l'ApplicationMaster de YARN.

Pour plus d'informations sur l'onglet **Hadoop Configuration**, consultez la section décrivant comment configurer la connexion Hadoop pour un Job **Talend** Map/Reduce dans le *Guide de prise en main de Talend Open Studio for Big Data*.

Pour plus d'informations sur le Resource Manager, son ordonnanceur et l'ApplicationMaster, consultez la documentation de YARN sur le site http://hortonworks.com/blog/apache-hadoop-yarn-concepts-and-applications/ (en anglais).

Pour plus d'informations concernant la définition des paramètres de mémoire pour YARN et MapReduce, consultez la documentation de la distribution utilisée comme, par exemple, celle d'Hortonworks : http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.0.6.0/bk_installing_manually_book/content/rpm-chap1-11.html (en anglais).

Configurer les composants

Configurer le tHDFSInput

1. Double-cliquez sur le **tHDFSInput** pour ouvrir sa vue **Component**.

2. Cliquez sur le bouton ... à côté du champ **Edit schema** afin de vérifier que le schéma reçu dans les étapes précédentes est bien défini.

Notez que si vous créer ce Job à partir de rien, vous devez cliquer sur le bouton afin de définir manuellement le schéma. Sinon, si le schéma est défini dans le **Repository**, vous pouvez sélectionner l'option **Repository** dans la liste **Schema**, dans la liste **Basic settings** afin de le réutiliser. Pour plus d'informations concernant la définition d'un schéma dans le **Repository**, consultez le chapitre concernant la gestion des métadonnées, dans le *Guide utilisateur du Studio Talend* ou le chapitre décrivant le nœud **Hadoop cluster** du **Repository**, dans le *Guide de prise en main de Talend Open Studio for Big Data*.

3. Si vous modifiez le schéma, cliquez sur **OK** afin de valider les modifications et acceptez la propagation proposée par la boîte de dialogue qui s'ouvre.

4. Dans le champ **Folder/File**, saisissez le chemin d'accès ou parcourez votre système jusqu'au fichier source devant être lu par le Job.

   Si ce fichier n'est pas dans le système HDFS à utiliser, vous devez le placer dans ce HDFS, à l'aide, par exemple, des composants **tFileInputDelimited** et **tHDFSOutput** dans un Job **Standard**.

Vérifier le composant de transformation

- Double-cliquez sur le **tConvertType** pour ouvrir la vue **Component**.

  Le composant conserve les paramètres des onglets **Basic settings** et **Advanced settings** utilisés dans le Job d'origine. Comme le fait l'original, il convertit les types **String** et **Float** en **Integer**.

Vérifier le tMap

- Double-cliquez sur le **tMap** pour ouvrir son éditeur. La configuration du mapping est la même que dans le Job d'origine, elle permet d'écrire en sortie la colonne *StringtoInteger* convertie et de calculer la somme des colonnes *IntegerField* et *FloatToInteger*.

Exécuter le Job

Vous pouvez exécuter ce Job.

Le composant **tLogRow** est utilisé pour présenter les résultats d'exécution du Job.

1. Si vous souhaitez configurer la présentation des résultats dans la vue **Component**, double-cliquez sur le **tLogRow** afin d'ouvrir sa vue **Component**. Dans la zone **Mode**, sélectionnez l'option **Table (print values in cells of a table)**.

2. Appuyez sur **F6** pour exécuter le Job.

Durant l'exécution, la vue **Run** s'ouvre automatiquement. Vous pouvez voir la progression du Job, ainsi que le statut du calcul Map/Reduce effectué par le Job.

Dans l'espace de modélisation graphique, une barre de progression apparaît automatiquement sous les composants effectuant des calculs Map/Reduce, afin d'afficher graphiquement le statut du calcul en cours.

```
Running job: job_201308211416_0095
 map 0% reduce 0%
 map 66% reduce 0%
 map 100% reduce 0%
Job complete: job_201308211416_0095
Counters: 25
    File System Counters
        FILE: Number of bytes read=0
        FILE: Number of bytes written=492444
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=329
        HDFS: Number of bytes written=6
        HDFS: Number of read operations=9
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=6
    Job Counters
        Launched map tasks=3
        Data-local map tasks=3
        Total time spent by all maps in occupied slots (ms)=19484
        Total time spent by all reduces in occupied slots (ms)=0
        Total time spent by all maps waiting after reserving slots (ms)=0
        Total time spent by all reduces waiting after reserving slots (ms)=0
    Map-Reduce Framework
        Map input records=1
        Map output records=1
        Input split bytes=306
        Spilled Records=0
        CPU time spent (ms)=2590
        Physical memory (bytes) snapshot=235491328
        Virtual memory (bytes) snapshot=1189625856
        Total committed heap usage (bytes)=150601728
    org.apache.hadoop.mapreduce.lib.input.FileInputFormatCounter
        BYTES_READ=13
.---------------+---.
|     tLogRow_1     |
|=---------------+--=|
|StringtoInteger|Sum|
|=---------------+--=|
|3              |579|
'---------------+---'
[statistics] disconnected

Job type_convert_mr terminé à 16:28 06/09/2013. [Code sortie=0]
```

Nombre limite de lignes 100 Retour automatique à la ligne

Si vous souhaitez plus d'informations concernant le Job et son exécution, utilisez la console Web fournie par la distribution Hadoop que vous utilisez.

tDenormalize

Propriétés du tDenormalize

Famille de composant	Processing/Fields	
Fonction	Le composant **tDenormalize** dénormalise un flux entrant en fonction d'une colonne.	
Objectif	Le **tDenormalize** permet de synthétiser un flux entrant.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant uniquement. Voir également le *Guide utilisateur du Studio Talend*.
	To denormalize	Ce tableau vous permet de choisir les paramètres de dénormalisation des colonnes. **Column** : Sélectionnez la colonne à dénormaliser. **Delimiter** : Saisissez, entre guillemets doubles, le séparateur que vous souhaitez utiliser. **Merge same value** : Cochez cette case si vous voulez fusionner les valeurs identiques.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n'est pas disponible dans la version Map/Reduce de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en étape intermédiaire dans un flux de données.
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Notez que ce composant peut changer l'ordre du flux Java entrant.

Scénario 1 : Dénormaliser une colonne

Ce scénario illustre un Job dénormalisant une colonne d'un fichier délimité via un composant de dénormalisation, et affichant le résultat dans la console.

1. Cliquez et déposez les composants suivants : **tFileInputDelimited**, **tDenormalize**, **tLogRow**.

2. Connectez ces composants à l'aide d'un lien **Row main**.

3. Dans la vue **Component** du composant **tFileInputDelimited**, renseignez le chemin d'accès au fichier à dénormaliser.

4. Dans les champs **Header**, **Row Separator** et **Field Separator**, renseignez respectivement le nombre de lignes d'en-tête, le type de séparateur de lignes et de champs.

5. Le fichier d'entrée est composé de deux colonnes : *Fathers* et *Children*.

6. Dans l'onglet **Basic settings** du **tDenormalize**, paramétrez la colonne contenant des valeurs multiples à regrouper. Dans le tableau **To denormalize**, sélectionnez la colonne *Children*.

7. Dans cet exemple, la colonne *Children* est celle à dénormaliser.

8. Dans le champ **Delimiter**, paramétrez le séparateur des valeurs regroupées. Attention, une seule colonne peut être dénormalisée.

9. Cochez la case **Merge same value** si vous souhaitez supprimer les éventuels doublons.

10. Enregistrez le Job et exécutez-le en appuyant sur **F6**.

```
Starting job tDenormalize at 10:50 09/03/2016.

Mickael|Océane
Stéphane|Agathe;Clémence
Pierrick|Erwann;Tiphaine
Robert|Manon
Richard|Roméo
Fabrice|Martin
Job tDenormalize ended at 10:50 09/03/2016. [exit code=0]
```

Toutes les valeurs de la colonne *Children* (définie comme colonne à dénormaliser) sont regroupées en fonction des valeurs de la colonne *Fathers*. Les valeurs sont séparées par des virgules.

Scénario 2 : Dénormaliser plusieurs colonnes

Ce scénario illustre un Job dénormalisant deux colonnes d'un fichier délimité via un composant de dénormalisation et affichant le résultat dans la console.

1. Cliquez et déposez les composants suivants : **tFileInputDelimited**, **tDenormalize**, **tLogRow**.

2. Connectez tous les composants à l'aide d'une connexion **Row** > **main**.

3. Dans l'onglet **Basic settings** du composant **tFileInputDelimited**, renseignez le chemin d'accès au fichier à dénormaliser.

4. Renseignez les autres champs, dont les champs **Row Separator**, **Field separators** et **Header** contenant respectivement les séparateurs de lignes et de champs et le nombre de lignes d'en-tête.

5. Le schéma du fichier est composé de quatre colonnes : *Name*, *FirstName*, *HomeCity*, *WorkCity*.

6. Dans l'onglet **Basic settings** du composant **tDenormalize**, sélectionnez les colonnes contenant les répétitions. Elles correspondent aux colonnes apparaissant plusieurs fois dans le document. Dans cet exemple, *FirstName*, *HomeCity* et *WorkCity* sont les colonnes à partir desquelles la dénormalisation est effectuée.

7. Ajoutez autant trois lignes à la table **To denormalize** que vous le souhaitez en cliquant sur le bouton **[+]**. Puis sélectionnez les colonnes *FirstName*, *HomeCity* et *WorkCity* dans la liste déroulante.

8. Dans la colonne **Delimiter**, paramétrez les délimiteurs en les saisissant entre guillemets, afin de séparer les valeurs concaténées. Pour la colonne *FirstName*, saisissez "#", pour la colonne *HomeCity*, saisissez "§", et pour la colonne *WorkCity*, saisissez "¤".

9. Enregistrez le Job et exécutez-le, en appuyant sur **F6**.

```
Starting job tDenormalize2 at 12:03 09/03/2010.

Moore|Demi#Demi|New York§Rio de Janeiro|Paris¤Los Angeles
Joli|Angelina#Angelina#Angelina|Berlin§Berlin§Los
Angeles|Berlin¤Los Angeles¤Los Angeles
Pitt|Brad#Brad|Beverly Hills§Paris|Los Angeles¤London
Willis|Bruce#Bruce#Bruce#Bruce|Paris§Paris§Madrid§Roma|Los
Angeles¤Madrid¤Madrid¤Dublin
Job tDenormalize2 ended at 12:03 09/03/2010. [exit code=0]
```

Le résultat affiche les valeurs dénormalisées concaténées à l'aide des séparateurs.

10. Dans l'onglet **Basic settings** du composant **tDenormalize**, dans le tableau **To denormalize**, cochez la case **Merge same value** pour fusionner les valeurs identiques.

11. Enregistrez le Job et exécutez-le en appuyant sur **F6**.

```
Starting job tDenormalize2 at 13:44 09/03/2010.

Moore|Demi|New York§Rio de Janeiro|Paris¤Los Angeles
Joli|Angelina|Berlin§Los Angeles|Berlin¤Los Angeles
Pitt|Brad|Beverly Hills§Paris|Los Angeles¤London
Willis|Bruce|Paris§Madrid§Roma|Los Angeles¤Madrid¤Dublin
Job tDenormalize2 ended at 13:44 09/03/2010. [exit code=0]
```

Cette fois-ci, la console affiche les résultats sans doublon.

tDenormalizeSortedRow

Propriétés du tDenormalizeSortedRow

Famille du composant	Processing/Fields	
Fonction	Le composant **tDenormalizeSortedRow** rassemble dans un groupe toutes les lignes d'entrée dénormalisées triées ainsi que leurs valeurs distinctes, associées par des séparateurs de champ.	
Objectif	Le **tDenormalizeSortedRow** permet de synthétiser les lignes d'entrée triées afin de faire des économies de mémoire.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Input rows count	Renseignez le nombre de lignes d'entrée.
	To denormalize	Renseignez le nom de la colonne à dénormaliser.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant traite des flux de données, il nécessite donc un composant d'entrée et un composant de sortie.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Regrouper des lignes triées

Ce scénario Java décrit un Job à quatre composants. Il permet de lire un fichier délimité donné ligne par ligne, de trier les données d'entrées en fonction de leur type et de leur ordre, de dénormaliser toutes les lignes d'entrées triées et enfin d'afficher le résultat dans la console de la vue **Run**.

- A partir de la **Palette**, cliquez-déposez les composants suivants dans l'éditeur graphique : le **tFileInputDelimited**, le **tSortRow**, le **tDenormalizeSortedRow** et le **tLogRow**.
- Connectez les quatre composants à l'aide de liens de type **Row Main**.

- Dans l'éditeur graphique, sélectionnez le composant **tFileInputDelimited**.
- Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tFileInputDelimited**.

- Définissez le champ **Property Type** en mode **Built-In**.

- Renseignez le chemin d'accès au fichier à traiter dans le champ **File Name**. Dans cet exemple, on utilise le fichier *name_list*, qui comporte deux colonnes, *id* et *first name*.

- Définissez si nécessaire les séparateurs de lignes et de champs, l'en-tête et le pied-de-page, ainsi que le nombre de lignes traitées.

- Définissez le champ **Schema** en mode **Built in** puis cliquez sur le bouton **[...]** à côté du champ **Edit Schema** pour définir les données à passer au composant suivant. Dans cet exemple, le schéma est constitué de deux colonnes, *id* et *name*.

- Dans l'espace graphique, sélectionnez le composant **tSortRow**.

- Cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**) du **tSortRow**.

- Définissez le champ **Schema Type** en mode **Built-In** puis cliquez sur **Sync columns** pour récupérer le schéma à partir du composant **tFileInputDelimited**.

- Dans le panneau **Criteria**, cliquez sur le bouton **[+]** pour ajouter une ligne puis définissez les paramètres de tri pour la colonne du schéma à traiter. Dans cet exemple, on veut trier la colonne *id* dans l'ordre croissant.

- Dans l'espace graphique, sélectionnez le composant **tDenormalizeSortedRow**.

- Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tDenormalizeSortedRow**.

- Définissez le champ **Schema** en mode **Built-In** puis cliquez sur **Sync columns** pour récupérer le schéma à partir du schéma du composant **tSortRow**.
- Dans le champ **Input rows count**, saisissez le nombre de lignes d'entrée à traiter ou cliquez simultanément sur **Ctrl+Espace** pour accéder à la liste des variables de contexte puis sélectionnez la variable : `tFileInputDelimited_1_NB_LINE`
- Dans le panneau **To denormalize**, cliquez sur le bouton **[+]** pour ajouter une ligne puis définissez les paramètres de la colonne à dénormaliser. Dans cet exemple, on veut dénormaliser la colonne *name*.
- Dans l'éditeur graphique, sélectionnez le composant **tLogRow** puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**). Pour plus d'informations sur le **tLogRow**, consultez *tLogRow*.
- Enregistrez votre Job puis appuyez sur **F6** pour l'exécuter.

```
Starting job denormalize1 at 14:42 01/09/2008.
bufferSize=100000 objects
Sorting buffer...
0 milliseconds for 9 objects to sort in memory. 2147483647
items/s
Writing ordered buffer in file...
16 milliseconds for 9 objects to write in file. 562  items/s
.--+-----------------------.
|        tLogRow_1          |
|=-+-----------------------=|
|id|name                    |
|=-+-----------------------=|
|1 | Harrison; Jerry        |
|2 | Ford                   |
|3 | James; Goldman         |
|4 | Bill; Ford; Harry; Sue|
'--+-----------------------'

Job denormalize1 ended at 14:42 01/09/2008. [exit code=0]
```

Le résultat affiché sur la console montre la façon dont la colonne *name* a été dénormalisée.

tExternalSortRow

Propriétés du tExternalSortRow

Famille du composant	Processing	
Fonction	Le composant **tExternalSortRow** utilise une application externe de tri pour trier les données d'entrée d'une ou plusieurs colonnes, en fonction de leur type et de leur ordre.	
Objectif	Le **tExternalSortRow** permet d'établir des métriques et des tables de classification.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	File Name	Chemin d'accès et nom du fichier à traiter, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Field separator	Caractère, chaîne ou expression régulière séparant les champs.
	External command "sort" path	Renseignez le chemin d'accès au fichier externe contenant l'algorithme de tri à utiliser.
	Criteria	Cliquez sur **[+]** pour ajouter autant de lignes que nécessaire pour mettre en place le tri. La première colonne définie dans le schéma est sélectionnée par défaut.
		Schema column : Sélectionnez la colonne de votre schéma sur laquelle vous souhaitez baser votre tri. Notez que l'ordre est important car il détermine la priorité de tri.
		Sort type : Numérique ou Alphabétique.
		Order : Ordre ascendant ou descendant.
Advanced settings	*Maximum memory*	Renseignez la taille de la mémoire physique que vous voulez allouer au processus de tri.

	Temporary directory	Définissez l'endroit où stocker les fichiers temporaires.
	Set temporary input file directory	Cochez la case pour afficher le champ dans lequel vous pouvez spécifier le répertoire du fichier d'entrée temporaire.
	Add a dummy EOF line	Cochez cette case lorsque vous utilisez le composant **tAggregateSortedRow**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant traite des flux de données, il nécessite donc un composant d'entrée et un composant de sortie.	

Scénario associé

Pour un scénario associé, consultez *tSortRow*.

tExtractDelimitedFields

Propriétés du tExtractDelimitedFields

Famille du composant	Processing/Fields	
Fonction	Le composant **tExtractDelimitedFields** génère des colonnes multiples à partir d'une colonne donnée dans un fichier délimité. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job Map/Reduce **Talend** afin de générer du code Map/Reduce. Pour plus d'informations, consultez *Le tExtractDelimitedFields dans des Jobs Map/Reduce Talend*.	
Objectif	Le **tExtractDelimitedFields** permet d'extraire les "champs" à partir d'une chaîne de caractères, pour les écrire à un autre endroit par exemple.	
Basic settings	*Field to split*	Sélectionnez un champ entrant à partir de la liste **Field to split**.
	Ignore NULL as the source data	Cochez cette case afin d'ignorer les valeurs Null dans les données source. Décochez cette case afin de générer les enregistrements Null correspondant aux valeurs Null dans les données source.
	Field separator	Définissez le séparateur de champs. La syntaxe regex utilise un certain nombre de caractères spéciaux comme opérateurs. Si vous utilisez l'un de ces opérateurs comme séparateur de champs, il est nécessaire de protéger le caractère par un double antislash. Utilisez par exemple "\\|" au lieu de "|".
	Die on error	Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*Advanced separator(for number)*	Cochez cette case pour modifier les séparateurs utilisés pour les nombres.
	Trim columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.
	Check each row structure against schema	Cochez cette case pour vérifier si le nombre total de colonnes pour chaque ligne correspond au schéma. S'il ne correspond pas, un message d'erreur s'affichera dans la console.
	Validate date	Cochez cette case pour vérifier strictement le format de la date par rapport au schéma d'entrée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant traite des flux de données, il nécessite donc un composant d'entrée et un composant de sortie. Il permet d'extraire des données d'un champ délimité à l'aide d'un lien **Row** > **Main**, et permet de créer un flux de rejet avec un lien **Row** > **Reject**, filtrant les données dont le type ne correspond pas au type défini.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Le tExtractDelimitedFields dans des Jobs Map/Reduce Talend

*Les informations contenues dans cette section concernent uniquement les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend** et ne sont pas applicables aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job Map/Reduce **Talend**, le **tExtractDelimitedFields**, ainsi que le Job complet utilisant ce composant, génère du code Map/Reduce natif. Cette section présente les propriétés spécifiques du **tExtractDelimitedFields** lorsqu'il est utilisé dans un Job Map/Reduce. Pour plus d'informations concernant un Job Map/Reduce **Talend**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	Processing / Fields	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Réutilisez les propriétés stockées centralement sous le nœud **Hadoop Cluster** de la vue **Repository**. Les champs suivants sont alors pré-remplis à l'aide des données collectées.

		Pour plus d'informations concernant le nœud **Hadoop Cluster**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.
	Field	Sélectionnez la colonne dans laquelle les champs doivent être divisés.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si vous effectuez des modifications, le schéma devient automatiquement built-in.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Die on error	Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	Field separator	Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.
	CSV options	Cochez cette case pour inclure des paramètres CSV spécifiques comme **Escape char** et **Text enclosure**.
Advanced settings	*Custom Encoding*	Il est possible de rencontrer des problèmes d'encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste **Encoding**. Sélectionnez l'encodage à utiliser dans la liste ou sélectionnez **Custom** pour le définir manuellement.
	Advanced separator (for number)	Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.).
	Trim all columns	Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées. Lorsque cette case est cochée, la table **Check column to trim** est affichée, ce qui vous permet de sélectionner les colonnes desquelles vous souhaitez enlever les espaces en début et en fin de champ.
	Check column to trim	Cette table est automatiquement renseignée avec le schéma utilisé. Cochez la (les) case(s) correspondant à la (aux) colonne(s) dont vous souhaitez supprimer les espaces et début et en fin de champ.
	Check each row structure against schema	Cochez cette case afin de vérifier que le nombre total de colonnes est cohérent par rapport au schéma. Si le nombre n'est pas cohérent, un message d'erreur s'affiche dans la console.
	Check date	Cochez cette case pour vérifier strictement le format de date par rapport au schéma d'entrée.
	Permit hexadecimal (0xNNN) or octal (0NNNN) for numeric types	Cochez cette case si l'un de vos types numériques (long, integer, short, ou byte), doit être parsé depuis une chaîne de caractères hexadécimale ou octale.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Lorsqu'un Job Map/Reduce est ouvert dans l'espace de modélisation graphique, le **tExtractDelimitedFields** et la famille **MapReduce** s'affichent dans la **Palette** du Studio. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.

Scénario : Extraire des champs à partir d'un fichier délimité par des virgules

Ce scénario Java décrit un Job à trois composants où le composant **tExtractDelimitedFields** est utilisé pour extraire deux colonnes à partir d'un fichier délimité par des virgules.

Relier les composants

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l'éditeur graphique : le **tFileInputDelimited**, le **tExtractDelimitedFields** et le **tLogRow**.

2. Effectuez un clic-droit sur chacun des trois composants pour les connecter à l'aide de liens **Row Main**.

Configurer les composants

1. Double-cliquez sur le **tFileInputDelimited** afin d'ouvrir sa vue **Basic settings**.

2. Dans l'onglet **Basic settings**, définissez le champ **Property Type** en mode **Built-In**.

3. Cliquez sur le bouton **[...]** à côté du champ **File Name** pour définir le chemin d'accès au fichier d'entrée.

   Le champ **File Name** est obligatoire.

   Dans ce scénario, on utilise le fichier d'entrée *test5*. Il s'agit d'un fichier texte dont certaines données sont séparées par des virgules.

4. Dans l'onglet **Basic settings**, renseignez les autres champs. Pour plus d'informations, consultez *tFileInputDelimited*. Dans ce scénario, l'en-tête et le pied-de-page ne sont pas définis et aucune limite n'est renseignée concernant le nombre de lignes traitées.

5. Cliquez sur **Edit schema** pour décrire la structure des données du fichier d'entrée. Dans ce scénario, le schéma est composé d'une seule colonne, *name*.

6. Double-cliquez sur le **tExtractDelimitedFields** afin d'ouvrir sa vue **Basic settings**.

7. A partir de la liste déroulante **Field to split**, sélectionnez la colonne à dissocier. Dans ce scénario, il s'agit de la colonne *name*.

8. Dans le champ **Field separator**, renseignez le séparateur de champs correspondant.

9. Cliquez sur **Edit schema** pour décrire la structure des données de ce composant de traitement.

10. Dans le panneau de sortie de la boîte de dialogue **[Schema of tExtractDelimitedFields]**, cliquez sur le bouton **[+]** pour ajouter deux colonnes, *firstname* et *lastname*, au schéma de sortie.

Dans ce scénario, on veut séparer la colonne *name* en deux colonnes dans le flux de sortie, *firstname* et *lastname*.

11. Cliquez sur **OK** pour fermer la boîte de dialogue **[Schema of tExtractDelimitedFields]**.

12. Dans l'espace graphique, sélectionnez le composant **tLogRow** puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**). Pour plus d'informations, consultez *tLogRow*.

Exécuter le Job

1. Appuyez sur **Ctrl + S** pour sauvegarder votre Job.

2. Appuyez sur **F6** pour l'exécuter.

```
Starting job extract_delimited at 16:39 25/08/2008.
.---------+--------.
|    tLogRow_1      |
|=--------+-------=|
|firstname|lastname|
|=--------+-------=|
|Janet    |Anderson|
|Martin   |Chairman|
|Lily     |Massy   |
'---------+--------'

Job extract_delimited ended at 16:39 25/08/2008. [exit
```

tExtractEBCDICFields

Propriétés du tExtractEBCDICFields

Famille de composant	Processing/Fields	Notez que ce composant est déprécié. Les capacités de traitement EBCDIC sont désormais entièrement supportées par *Talend Data Mapper*, qui est inclus avec n'importe quelle solution de **Talend** Platform. Pour plus d'informations concernant *Talend Data Mapper*, consultez le *Guide utilisateur de Talend Data Mapper*.
Fonction	Le composant **tExtractEBCDICFields** génère de multiples colonnes à partir d'une colonne, à l'aide du schéma EBCDIC défini.	
Objectif	Le **tExtractEBCDICFields** vous permet d'extraire des données EBCDIC d'une chaîne de caractères formatée.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Sélectionnez cette option pour éditer le schéma Built-in pour les données à traiter.
		Repository : Sélectionnez cette option pour éditer le schéma du **Repository** sélectionné. Les champs suivants sont automatiquement renseignés à l'aide du schéma sélectionné..
	Sync columns	Cliquez sur ce bouton pour récupérer le schéma défini dans le composant d'entrée. Ce bouton est disponible uniquement lorsqu'un composant d'entrée est connecté à ce composant via un lien **Row** > **Main**.
	Field	Sélectionnez dans la liste **Field** un champ entrant à extraire.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Encoding*	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

	Trim all column	Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées. Lorsque cette case est cochée, la table **Check column to trim** est affichée, ce qui vous permet de sélectionner les colonnes desquelles vous souhaitez enlever les espaces en début et en fin de champ.
	Check column to trim	Cette table est automatiquement renseignée avec le schéma utilisé. Cochez la (les) case(s) correspondant à la (aux) colonne(s) dont vous souhaitez supprimer les espaces et début et en fin de champ.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant gère un flux de données et nécessite donc un composant d'entrée et un composant de sortie. Cela vous permet d'extraire des données d'un fichier délimité, à l'aide d'un lien **Row** > **Main** et vous permet de créer un flux de rejet filtrant des données dont le type ne correspond pas au type défini.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario

Pour un scénario associé, consultez *Scénario : Extraire les données d'un fichier EBCDIC*.

tExtractJSONFields

Propriétés du tExtractJSONFields

Famille de composant	JSON/Processing	
Fonction	Le composant **tExtractJSONFields** extrait les données souhaitées provenant des champs JSON d'entrée, selon la requête XPath ou JSONPath.	
Objectif	Le **tExtractJSONFields** extrait les données des champs JSON stockés dans un fichier, une table de base de données, etc., selon la requête XPath ou JSONPath.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit **Built-in** soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Read By	Sélectionnez dans la liste une manière d'extraire le champ JSON spécifié : **Xpath** ou **JsonPath**.
	JSON field	Liste des champs JSON à extraire.
	Loop XPath query	Nœud, dans le champ JSON, sur lequel se base la boucle. Ce champ est disponible uniquement lorsque l'option **Xpath** est sélectionnée dans la liste **Read By**.
	Loop Jsonpath query	Spécifiez le nœud JSONPath sur lequel se base la boucle. Ce champ est disponible uniquement lorsque l'option **JsonPath** est sélectionnée dans la liste **Read By**.
	Mapping	Renseignez cette table afin de mapper les colonnes définies dans le schéma aux nœuds JSON correspondants. • **Column** : les cellules **Column** sont automatiquement renseignées avec les noms des colonnes du schéma.

		• **XPath Query** : spécifiez le nœud XPath contenant les données souhaitées. Cette colonne est disponible uniquement lorsque l'option **Xpath** est sélectionnée dans la liste **Read By**. • **Get nodes** : cochez cette case pour extraire les données JSON de tous les nœuds spécifiés dans la liste **Xpath query** ou cochez la case à côté d'un nœud spécifique pour extraire les données de ce nœud uniquement. Cette colonne est disponible uniquement lorsque l'option **Xpath** est sélectionnée dans la liste **Read By**. • **Is Array** : cochez cette case lorsque le champ JSON à extraire est un tableau et non un objet. Cette colonne est disponible uniquement lorsque l'option **Xpath** est sélectionnée dans la liste **Read By**. • **Json query** : spécifiez le nœud JSONPath contenant les données souhaitées. Pour plus d'informations concernant les expressions XPath, consultez http://goessner.net/articles/JsonPath/. Cette colonne est disponible uniquement lorsque l'option **JsonPath** est sélectionnée dans la liste **Read By**.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
Advanced settings	*Encoding*	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant. Notez que cette case n'est pas disponible dans les versions Map/Reduce et Storm de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant intermédiaire. Il nécessite un composant d'entrée et un composant de sortie.	
Utilisation dans les Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet.	

	Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Utilisation dans des Jobs Storm	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez également utiliser ce composant en tant que composant Storm. Dans un Job Storm **Talend**, ce composant est utilisé comme étape intermédiaire. Les autres composants utilisés avec celui-ci doivent également être des composants Storm. Ils génèrent nativement du code Storm pouvant être exécuté directement dans un système Storm. La version de Storm ne supporte pas l'utilisation des variables globales. Vous devez utiliser l'onglet **Storm Configuration** dans la vue **Run** pour définir la connexion à un système Storm donné pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant un Job Storm **Talend**, consultez les sections décrivant comment créer et configurer un Job Storm **Talend** dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf lorsqu'explicitement indiqué, un scénario présente uniquement des Jobs de type **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.

Scénario : Récupérer les messages d'erreur lors de l'extraction de données de champs JSON

Dans ce scénario, le **tWriteJSONField** transforme les données d'entrée en des champs JSON, données ensuite extraites par le **tExtractJSONFields**. Les messages d'erreur générés par les échecs de l'extraction, notamment les champs et erreurs JSON concernés, sont récupérés via un lien **Row** > **Reject**.

Construire le Job

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tFixedFlowInput**, un **tWriteJSONField**, un **tExtractJSONFields** et deux **tLogRow**. Renommez les deux **tLogRow**, respectivement *data_extracted* et *reject_info*.
2. Reliez le **tFixedFlowInput** au **tWriteJSONField** à l'aide d'un lien **Row** > **Main**.
3. Connectez le composant **tWriteJSONField** au **tExtractJSONFields** à l'aide d'un lien **Row** > **Main**.
4. Reliez le **tExtractJSONFields** au **data_extracted** à l'aide d'un lien **Row** > **Main**.

5. Reliez le **tExtractJSONFields** au **reject_info** à l'aide d'un lien **Row** > **Reject**.

Configurer les composants

Configurer le tFixedFlowInput

1. Double-cliquez sur le **tFixedFlowInput** pour afficher sa vue **Basic settings**.

2. Cliquez sur **Edit schema** pour ouvrir l'éditeur de schéma.

Cliquez trois fois sur le bouton **[+]** pour ajouter trois colonnes, respectivement *firstname*, *lastname* et *dept*, de type string.

Cliquez sur **OK** pour fermer l'éditeur.

3. Sélectionnez **Use Inline Content** et saisissez les données ci-dessous dans le champ **Content** :

```
Andrew;Wallace;Doc
John;Smith;R&D
Christian;Dior;Sales
```

Configurer le tWriteJSONField

1. Cliquez sur le composant **tWriteJSONField** pour afficher sa vue **Basic settings**.

2. Cliquez sur **Configure JSON Tree** pour ouvrir l'éditeur d'arborescence XML.

Le schéma du **tFixedFlowInput** apparaît dans le panneau **Linker source**.

3. Dans le panneau **Linker target**, cliquez sur le **rootTag** par défaut et saisissez **staff**, le nœud racine du champ JSON à générer.

4. Cliquez-droit sur **staff** et sélectionnez **Add Sub-element** dans le menu contextuel.

5. Dans la boîte de dialogue qui s'ouvre, saisissez le nom du sous-nœud, *firstname*.

Répétez ces étapes pour ajouter deux autres sous-nœuds, respectivement *lastname* et *dept*.

6. Cliquez-droit sur **firstname** et sélectionnez **Set As Loop Element** dans le menu contextuel.
7. Déposez la ligne **firstname** du panneau **Linker source** dans son homonyme du panneau **Linker target**.

   Dans la boîte de dialogue qui s'ouvre, sélectionnez **Add linker to target node**.

   Cliquez sur **OK** pour fermer la boîte de dialogue.

8. Répétez les étapes pour relier les deux autres éléments.

   Cliquez sur **OK** pour fermer l'éditeur d'arborescence XML.

9. Cliquez sur **Edit schema** pour ouvrir l'éditeur de schéma.

10. Cliquez sur le bouton **[+]** dans le panneau de droite pour ajouter une colonne, nommez-la *staff*. La colonne contiendra les données JSON générées.

    Cliquez sur **OK** pour fermer l'éditeur.

Configurer le tExtractJSONFields

1. Double-cliquez sur le **tExtractJSONFields** pour afficher sur la vue **Basic settings**.

2. Cliquez sur **Edit schema** pour ouvrir l'éditeur du schéma.

3. Cliquez trois fois sur le bouton **[+]** dans le panneau droit pour ajouter trois colonnes, respectivement *firstname*, *lastname* et *dept*, qui contiendront les données de leur nœud homonyme dans le champ JSON *staff*.

   Cliquez sur **OK** pour fermer l'éditeur.

4. Dans la boîte de dialogue **[Propagate]**, cliquez sur **Yes** pour propager le schéma aux composants suivants.

5. Dans le champ **Loop XPath query**, saisissez *"/staff"*, le nœud racine des données JSON.

6. Dans la zone **Mapping**, saisissez le nom du nœud des données JSON dans la colonne **XPath query**. Les données de ces nœuds seront extraites et passées à leurs colonnes homonymes définies dans le schéma de sortie.

7. Définissez la requête XPath *"firstname"* pour la colonne *firstname*, *"lastname"* pour la colonne *lastname* et *""* pour la colonne *dept*. Notez que *""* n'est pas une requête XPath valide et conduit à des erreurs d'exécution.

Configurer les composants tLogRow

1. Double-cliquez sur le composant **data_extracted** pour afficher sa vue **Basic settings**.

2. Sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

3. Effectuez la même étape pour l'autre composant **tLogRow**, **reject_info**.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.

2. Cliquez sur **F6** pour exécuter le Job.

```
+-----------------------------------------------------------------------------------------------.
|                                                                reject_info
|
|=--------+---------+----+-------------------------------------------------------------------+---------
+-----------------------------------------------------------------------------------------------=|
|firstname|lastname|dept|errorJSONField                                                      |errorCode|errorMessage
|
|=--------+---------+----+-------------------------------------------------------------------+---------
+-----------------------------------------------------------------------------------------------=|
|Andrew   |Wallace |null|{"staff":{"firstname":"Andrew","lastname":"Wallace","dept":"Doc"}}  |null     |Invalid XPath expression:  Unexpected
'' - Line: 0|
|John     |Smith   |null|{"staff":{"firstname":"John","lastname":"Smith","dept":"R&D"}}      |null     |Invalid XPath expression:  Unexpected
'' - Line: 1|
|Christian|Dior    |null|{"staff":{"firstname":"Christian","lastname":"Dior","dept":"Sales"}}|null     |Invalid XPath expression:  Unexpected
'' - Line: 2|
'---------+---------+----+-------------------------------------------------------------------+---------
+-----------------------------------------------------------------------------------------------'
.---------+---------+----.
|    data_extracted     |
|=--------+---------+----=|
|firstname|lastname|dept|
|=--------+---------+----=|
'---------+---------+----'
```

Le lien de rejet montre des détails, comme les données extraites, les champs JSON dont les données ne sont pas extraites et causent l'échec de l'extraction.

Scénario 2 : Collecter des données de votre réseau social favori

Dans ce scénario, le **tFileInputJSON** récupère le nœud *friends* d'un fichier JSON contenant les données d'un utilisateur Facebook et le **tExtractJSONFields** extrait les données du nœud *friends* afin d'obtenir des données plates en sortie.

Relier les composants

1. Déposez les composants suivantes de la **Palette** dans l'espace de modélisation graphique : un **tFileInputJSON**, un **tExtractJSONFields** et un **tLogRow**.

2. Reliez le **tFileInputJSON** au **tExtractJSONFields** à l'aide d'un lien **Row** > **Main**.

3. Reliez le **tExtractJSONFields** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFileInputJSON** pour afficher sa vue **Basic settings**.

2. Cliquez sur **Edit schema** pour ouvrir l'éditeur de schéma.

Cliquez sur le bouton **[+]** pour ajouter une colonne, que vous nommez *friends*, de type String.

Cliquez sur **OK** pour fermer l'éditeur.

3. Cliquez sur le bouton **[...]** pour parcourir votre système jusqu'au fichier JSON, *facebook.json* dans cet exemple :

```
{ "user": { "id": "9999912398",
            "name": "Kelly Clarkson",
            "friends": [
                 { "name": "Tom Cruise",
                   "id": "55555555555555",
                   "likes": {
                       "data": [
                            { "category": "Movie",
                              "name": "The Shawshank Redemption",
                              "id": "103636093053996",
                              "created_time": "2012-11-20T15:52:07+0000"
                            },
                            { "category": "Community",
                              "name": "Positiveretribution",
                              "id": "471389562899413",
                              "created_time": "2012-12-16T21:13:26+0000"
                            }
                                ]
                            }
                 },
                 { "name": "Tom Hanks",
                   "id": "88888888888888"
                   "likes": {
                        "data": [
                            { "category": "Journalist",
                              "name": "Janelle Wang",
                              "id": "136009823148851",
                              "created_time": "2013-01-01T08:22:17+0000"
                            },
                            { "category": "Tv show",
                              "name": "Now With Alex Wagner",
                              "id": "305948749433410",
                              "created_time": "2012-11-20T06:14:10+0000"
                            }
                            ]
                           }
                  }
                        ]
           }
}
```

4. Décochez la case **Read by XPath**.

Dans la table **Mapping**, saisissez la requête JSONPath *"$.user.friends[*]"*, dans la colonne **JSONPath query**, pour la ligne *friends*, permettant de récupérer le nœud *friends* complet du fichier source.

5. Double-cliquez sur le **tExtractJSONFields** pour afficher sa vue **Basic settings**.

6. Cliquez sur **Edit schema** pour ouvrir l'éditeur de schéma.

7. Cliquez cinq fois sur le bouton **[+]** dans le panneau de droite pour ajouter cinq colonnes. Nommez-les respectivement *id*, *name*, *like_id*, *like_name* et *like_category*. Chaque colonne contiendra les données des nœuds concernés dans le champ JSON *friends*.

   Cliquez sur **OK** pour fermer l'éditeur.

8. Dans la boîte de dialogue **[Propagate]** qui s'ouvre, cliquez sur **Yes** pour propager le schéma aux composants suivants.

9. Dans le champ **Loop XPath query**, saisissez *"/likes/data"*.

10. Dans la zone **Mapping**, saisissez les requêtes des nœuds JSON dans la colonne **XPath query**. Les données de ces nœuds seront extraites et passées aux colonnes du même nom, dans le schéma de sortie.

11. Configurez la requête XPath *"../../id"* (interrogeant le nœud *"/friends/id"*) pour la colonne *id*.

    Configurez également la requête *"../../name"* (interrogeant le nœud *"/friends/name"*) pour la colonne *name*, ainsi que les requêtes *"id"* pour la colonne *like_id*, *"name"* pour la colonne *like_name* et *"category"* pour la colonne *like_category*.

12. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.

13. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.

2. Cliquez sur **F6** pour exécuter le Job.

```
.---------------+-----------+----------------+-------------------------+-------------.
|                                   tLogRow_1                                         |
|=--------------+-----------+----------------+-------------------------+-------------=|
|id             |name       |like_id         |like_name                |like_category|
|=--------------+-----------+----------------+-------------------------+-------------=|
|55555555555555|Tom Cruise|103636093053996|The Shawshank Redemption|Movie        |
|55555555555555|Tom Cruise|471389562899413|Positiveretribution     |Community    |
|88888888888888|Tom Hanks |136009823148851|Janelle Wang            |Journalist   |
|88888888888888|Tom Hanks |305948749433410|Now With Alex Wagner    |Tv show      |
'               '           '                '                         '             '
```

Comme affiché ci-dessus, les données relatives aux amis de l'utilisateur Facebook Kelly Clarkson sont correctement extraites.

tExtractPositionalFields

Propriétés du tExtractPositionalFields

Famille du composant	Processing/Fields	
Fonction	Le composant **tExtractPositionalFields** génère des colonnes multiples à partir d'une colonne donnée en utilisant des champs positionnels.	
Objectif	Le **tExtractPositionalFields** permet d'extraire les données d'un fichier positionnel selon un modèle (*pattern*).	
Basic settings	*Field*	Sélectionnez dans la liste **Field** le champ entrant à extraire. Dans la version Map/Reduce de ce composant, le nom du champ est **Prev.Comp.Column list**, ce qui indique la liste de la colonne du composant précédent.
	Ignore NULL as the source data	Cochez cette case afin d'ignorer les valeurs Null dans les données source. Décochez cette case afin de générer les enregistrements Null correspondant aux valeurs Null dans les données source. Notez que cette fonctionnalité n'est pas disponible dans la version Map/Reduce.
	Customize	Cochez cette case pour personnaliser le format des données du fichier positionnel et renseignez les colonnes du tableau **Formats**. **Column** : Sélectionnez la colonne que vous souhaitez personnaliser. **Size** : Saisissez la taille correspondant à la colonne. **Padding char** : saisissez, entre guillemets, le caractère de remplissage à supprimer du champ. Le caractère par défaut est un espace. **Alignment** : Sélectionnez le paramètre d'alignement approprié.
	Pattern	Renseignez le modèle à utiliser pour l'extraction. Un modèle est constitué de valeurs de longueur séparées par des virgules et interprétées comme une chaîne de caractères entre guillemets. Assurez-vous que les valeurs saisies dans ces champs correspondent au schéma défini.
	Die on error	Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Property type	Peut être **Built-in** ou **Repository**. Cette fonctionnalité est disponible uniquement dans la version Map/Reduce.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Réutilisez les propriétés stockées centralement sous le nœud **Hadoop Cluster** de la vue **Repository**. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Pour plus d'informations concernant le nœud **Hadoop Cluster**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.
Advanced settings	*Advanced separator(for number)*	Cochez cette option pour modifier les séparateurs utilisés pour les nombres.
	Trim Column	Cochez cette case afin de supprimer les espaces en début et en fin de champ, dans toutes les colonnes.
	Check each row structure against schema	Cochez cette case afin de vérifier que le nombre total de colonnes est cohérent par rapport au schéma. Si le nombre n'est pas cohérent, un message d'erreur s'affiche dans la console.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. La variable **NB_LINE** n'est pas disponible en version Map/Reduce. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant traite des flux de données, il nécessite donc un composant d'entrée et un composant de sortie. Il permet d'extraire des données d'un champ délimité à l'aide d'un lien **Row** > **Main**, et permet de créer un flux de rejet avec un lien **Row** > **Reject**, filtrant les données dont le type ne correspond pas au type défini.	
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet.	

	Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour un scénario associé, consultez *Scénario : Extraire des noms, des domaines et domaines de premier niveau à partir d'adresses e-mail*.

tExtractRegexFields

Propriétés du tExtractRegexFields

Famille du composant	Processing/Fields	
Fonction	Le composant **tExtractRegexFields** génère des colonnes multiples à partir d'une colonne donnée en utilisant des correspondances regex.	
Objectif	Le **tExtractRegexFields** permet d'utiliser des expressions régulières pour extraire les données selon un modèle.	
Basic settings	*Field to split*	Sélectionnez un champ entrant à diviser dans la liste **Field to split**.
	Regex	Renseignez l'expression régulière en fonction du langage de programmation que vous utilisez.
	Property type	Peut être **Built-in** ou **Repository**. Cette fonctionnalité est disponible uniquement dans la version Map/Reduce.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Réutilisez les propriétés stockées centralement sous le nœud **Hadoop Cluster** de la vue **Repository**. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Pour plus d'informations concernant le nœud **Hadoop Cluster**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent. *Vérifiez que le schéma de sortie ne contient pas de colonne ayant le même nom que la colonne d'entrée à diviser. Sinon, l'expression régulière ne fonctionnera pas comme supposé.*
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.

Advanced settings	*Die on error*	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
	Check each row structure against schema	Cochez cette case afin de vérifier que le nombre total de colonnes est cohérent par rapport au schéma. Si le nombre n'est pas cohérent, un message d'erreur s'affiche dans la console. Cette fonctionnalité n'est pas disponible en version Map/Reduce du **tExtractRegexFields**.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données. Cette fonctionnalité est disponible uniquement en version Map/Reduce.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant traite des flux de données, il nécessite donc un composant d'entrée et un composant de sortie. Il permet d'extraire des données d'un champ délimité à l'aide d'un lien **Row** > **Main**, et permet de créer un flux de rejet avec un lien **Row** > **Reject**, filtrant les données dont le type ne correspond pas au type défini.	
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Extraire des noms, des domaines et domaines de premier niveau à partir d'adresses e-mail

Ce scénario décrit un Job à trois composants dans lequel le **tExtractRegexFields** est utilisé pour définir une expression régulière correspondant à l'une des colonnes des données d'entrée. Dans ce scénario, il s'agit de la colonne *email*. Le composant **tExtractRegexFields** est également utilisé pour trouver les correspondances avec ladite expression régulière. Cette expression comprend des identifieurs de champ permettant de séparer le nom d'utilisateur, le nom de domaine ainsi que le nom du domaine de premier niveau pour chaque adresse e-mail. Si l'adresse e-mail est valide, ces trois parties sont extraites et affichées sur la console dans trois colonnes différentes. Les données des deux autres colonnes d'entrée, *id* et *age*, sont également extraites puis redirigées.

Construire le Job

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l'espace graphique : le **tFileInputDelimited**, **le tExtractRegexFields** et le **tLogRow**.
2. Connectez le **tFileInputDelimited** au **tExtractRegexFields** à l'aide d'un lien **Row** > **Main**, puis reliez le **tExtractRegexFields** au **tLogRow** via un autre lien **Row** > **Main**.

Configurer les composants

1. Dans l'espace graphique, double-cliquez sur le composant **tFileInputDelimited** pour définir sa configuration de base.

2. Cliquez sur le bouton **[...]** à côté du champ **File Name/Stream**pour définir le chemin d'accès au fichier d'entrée.

   Dans ce scénario, le fichier d'entrée utilisé est *test4*. Il s'agit d'un fichier texte comportant trois colonnes : *id*, *email* et *age*.

```
id;email;age
1;anna@yahoo.net;24
2;diana@sohu.com;31
3;fiona@gmail.org;20
```

Pour plus d'informations, consultez *tFileInputDelimited*.

3. Cliquez sur le bouton **[...]** du champ **Edit schema** pour définir la structure des données du fichier d'entrée.

4. Dans l'espace graphique, double-cliquez sur le composant **tExtractRegexFields** pour définir sa configuration de base (**Basic settings**).

5. A partir de la liste déroulante **Field to split**, sélectionnez la colonne à séparer, à savoir *email* dans ce scénario.

6. Dans le panneau **Regex**, renseignez l'expression régulière à utiliser pour effectuer la correspondance. Dans ce scénario, une expression régulière `"([a-z]*)@([a-z]*).([a-z]*)"` est utilisée pour mettre en correspondance les trois parties d'une adresse e-mail : utilisateur, domaine et domaine de premier niveau.

   Pour plus d'informations concernant les expressions régulières consultez http://fr.wikipedia.org/wiki/Expression_rationnelle.

7. Cliquez sur **Edit schema** pour ouvrir la boîte de dialogue **[Schema of tExtractRegexFields]**, cliquez cinq fois sur le bouton **[+]** pour ajouter cinq colonnes au schéma.

   Dans ce scénario, vous allez diviser la colonne d'entrée *email* en trois colonnes dans le flux de sortie, *name*, *domain* et *tld*. Les deux autres colonnes d'entrée seront extraites telles qu'elles sont.

8. Double-cliquez sur le composant **tLogRow** pour ouvrir sa vue **Component**.

9. Dans la zone Mode, sélectionnez **Table (print values in cells of a table)**.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour enregistrer votre Job.
2. Appuyez sur **F6** pour l'exécuter, ou cliquez sur l'onglet **Run**, dans la vue **Run**.

Le composant **tExtractRegexFields** effectue une correspondance entre toutes les adresses e-mail données et l'expression régulière définie, puis extrait le nom d'utilisateur, le domaine et le domaine de premier niveau, qu'il affiche sur la console dans trois colonnes séparées. Les deux autres colonnes, *id* et *age*, sont extraites telles quelles.

tExtractXMLField

tExtractXMLField appartient à deux familles de composants : Processing et XML. Pour plus d'informations concernant le composant **tExtractXMLField**, consultez *tExtractXMLField*.

tFilterColumns

Propriétés du tFilterColumns

Famille du composant	Processing	
Fonction	Le composant **tFilterColumns** opère des modifications spécifiques, établies à partir d'un mapping du nom des colonnes, sur un schéma défini.	
Objectif	Le **tFilterColumns** permet d'homogénéiser des schémas, soit en fonction de l'ordre des colonnes, soit en supprimant ou en ajoutant des colonnes.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository** Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant. Notez que cette case n'est pas disponible dans les versions Map/Reduce et Storm de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.	

	Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant n'est pas un composant de début (fond vert) et il nécessite un composant de sortie.
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Utilisation dans des Jobs Storm	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez également utiliser ce composant en tant que composant Storm. Dans un Job Storm **Talend**, ce composant est utilisé comme étape intermédiaire. Les autres composants utilisés avec celui-ci doivent également être des composants Storm. Ils génèrent nativement du code Storm pouvant être exécuté directement dans un système Storm. La version de Storm ne supporte pas l'utilisation des variables globales. Vous devez utiliser l'onglet **Storm Configuration** dans la vue **Run** pour définir la connexion à un système Storm donné pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant un Job Storm **Talend**, consultez les sections décrivant comment créer et configurer un Job Storm **Talend** dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf lorsqu'explicitement indiqué, un scénario présente uniquement des Jobs de type **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario associé

Pour plus d'informations concernant l'utilisation du composant **tFilterColumns**, consultez *Scénario : Remplacement multiple et filtrage de colonnes* du composant **tReplace**.

tFilterRow

Propriétés du tFilterRow

Famille du composant	Processing	
Fonction	Le composant **tFilterRow** filtre des lignes d'entrée en définissant une ou plusieurs condition(s) sur les colonnes sélectionnées.	
Objectif	Le composant **tFilterRow** aide à paramétrer les filtres sur les données source.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule.
	Logical operator used to combine conditions	Sélectionnez un opérateur logique afin de combiner des conditions simples et de combiner les résultats de filtre des deux modes si des conditions avancées ont été définies. **And** : retourne la valeur booléenne *true* si toutes les conditions sont vraies. Sinon, retourne *false*. Pour chacune des deux conditions combinées à l'aide de l'opérateur logique AND, la seconde condition est évaluée uniquement si la première est *true*. **Or** : retourne la valeur booléenne *true* si une condition est vraie. Sinon, retourne la valeur *false*. Pour les deux conditions combinées à l'aide de l'opérateur logique OR, la seconde condition est évaluée uniquement si la première est *false*.
	Conditions	Cliquez sur le bouton **[+]** pour ajouter autant de conditions simples que nécessaire. Selon l'opérateur logique sélectionné, les conditions sont évaluées les unes après les autres en ordre séquentiel pour chaque ligne. Lorsqu'elles sont évaluées, les conditions retournent la valeur booléenne *true* ou *false*. **Input column** : Sélectionnez la colonne du schéma sur laquelle exécuter la fonction. **Function** : Sélectionnez une fonction dans la liste. **Operator** : Sélectionnez l'opérateur permettant de relier la colonne d'entrée et la valeur. **Value** : Renseignez la valeur filtrée, entre guillemets si besoin est.
	Use advanced mode	Cochez cette case lorsque les opérations à effectuer ne peuvent aboutir via les fonctions standard proposées, par exemple différentes opérations logiques dans le même composant. Dans le champ texte, saisissez l'expression régulière comme demandé. Si plusieurs conditions avancées sont définies, utilisez un opérateur logique entre deux conditions : `&&` (opérateur logique AND) : retourne la valeur booléenne *true* si les deux conditions sont vraies, sinon retourne *false*. La seconde condition est évaluée uniquement si la première condition est *true*. `||` (opérateur logique OR) : retourne la valeur booléenne *true* si l'une des conditions est vraie, sinon, retourne *false*. La seconde condition est évaluée uniquement si la première est *false*.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n'est pas disponible dans les versions Map/Reduce et Storm de ce composant.

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **NB_LINE_OK** : nombre de lignes correspondant au filtre. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant n'est pas un composant de début (fond vert) et il nécessite un composant de sortie.
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Utilisation dans des Jobs Storm	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez également utiliser ce composant en tant que composant Storm. Dans un Job Storm **Talend**, ce composant est utilisé comme étape intermédiaire. Les autres composants utilisés avec celui-ci doivent également être des composants Storm. Ils génèrent nativement du code Storm pouvant être exécuté directement dans un système Storm. La version de Storm ne supporte pas l'utilisation des variables globales. Vous devez utiliser l'onglet **Storm Configuration** dans la vue **Run** pour définir la connexion à un système Storm donné pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant un Job Storm **Talend**, consultez les sections décrivant comment créer et configurer un Job Storm **Talend** dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf lorsqu'explicitement indiqué, un scénario présente uniquement des Jobs de type **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario 1 : Filtrer une liste de noms à l'aide de conditions simples

Le scénario suivant décrit un Job qui utilise des conditions simples pour filtrer une liste d'enregistrements. Le Job écrit deux tables de sortie : la première liste toutes les personnes de sexe masculin dont le nom de famille contient moins de neuf caractères et âgées de 10 à 80 ans. La seconde liste tous les enregistrements rejetés. Un message d'erreur s'affichera dans la table pour chaque enregistrement rejeté, afin d'expliquer pourquoi l'enregistrement a été rejeté.

Construire le Job

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l'éditeur graphique : un **tFixedFlowInput**, un **tFilterRow** et un **tLogRow**.

2. Connectez le **tFixedFlowInput** au **tFilterRow** à l'aide d'un lien **Row > Main**. Ensuite, connectez le **tFilterRow** au **tLogRow**, à l'aide d'un lien **Row > Filter**.

3. Déposez de la **Palette** dans l'espace de modélisation graphique un nouveau **tLogRow** et renommez-le *reject*. Reliez le **tFilterRow** à ce **tLogRow** à l'aide d'un lien **Row** > **Reject**.

4. Renommez les autres composants afin de mieux identifier leur rôle au sein du Job.

Configurer les composants

1. Dans l'espace graphique, cliquez sur le composant **tFixedFlowInput** afin d'afficher sa vue **Basic settings** et définir ses propriétés.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin de définir le schéma des données d'entrée. Dans cet exemple, le schéma se composant des quatre colonnes suivantes : *LastName* (type String), *Gender* (type String), *Age* (type Integer) et *City* (type String).

Cela fait, cliquez sur **OK** afin de valider les paramètres du schéma et fermer la boîte de dialogue. Une nouvelle boîte de dialogue s'ouvre et vous propose de propager le schéma. Cliquez sur **Yes**.

3. Dans les champs **Row separator** et **Field separator**, saisissez respectivement le séparateur de lignes et le séparateur de champs, si nécessaire. Dans cet exemple, utilisez les paramètres par défaut pour les deux champs, respectivement un retour chariot et un point-virgule.

4. Sélectionnez l'option **Use Inline Content(delimited file)** dans la zone **Mode** et saisissez les données d'entrée dans le champ **Content**.

Les données d'entrée utilisées dans cet exemple sont les suivantes :

```
Van Buren;M;73;Chicago
Adams;M;40;Albany
Jefferson;F;66;New York
Adams;M;9;Albany
Jefferson;M;30;Chicago
Carter;F;26;Chicago
Harrison;M;40;New York
```

```
Roosevelt;F;15;Chicago
Monroe;M;8;Boston
Arthur;M;20;Albany
Pierce;M;18;New York
Quincy;F;83;Albany
McKinley;M;70;Boston
Coolidge;M;4;Chicago
Monroe;M;60;Chicago
```

5. Double-cliquez sur le composant **tFilterRow** pour afficher sa vue **Basic settings** et définir ses propriétés.

6. Dans le tableau **Conditions**, ajoutez quatre conditions et renseignez les paramètres de filtre.

   - Pour la première ligne, dans la colonne **InputColumn**, dans la première ligne, sélectionnez *LastName*. Dans la colonne **Function**, sélectionnez **Length**. Dans la colonne **Operator**, sélectionnez **Lower than** et, dans la colonne **Value**, saisissez *9* afin de limiter la longueur les noms de famille à 9 caractères.

   - Pour la deuxième ligne, dans la colonne **InputColumn**, sélectionnez *Gender*. Dans la colonne **Operator**, sélectionnez **Equals** et, dans la colonne **Value**, saisissez *M* entre guillemets doubles afin de filtrer les enregistrement des personnes de sexe masculin.

     *Dans la colonne **Value**, vous devez saisir vos valeurs entre guillemets doubles pour tous les types de valeurs, sauf pour les valeurs de type Integer, qui ne nécessitent pas de guillemet.*

   - Dans la troisième ligne, dans la colonne **InputColumn**, sélectionnez *Age*. Dans la colonne **Operator**, sélectionnez **Greater than** et, dans la colonne **Value**, saisissez *10* afin de configurer l'âge minimum à 10 ans.

   - Dans la dernière ligne, dans la colonne **InputColumn**, sélectionnez *Age*. Dans la colonne **Operator**, sélectionnez **Lower than** et, dans la colonne **Value** saisissez *80* pour configurer l'âge maximum à 80 ans.

7. Pour combiner les deux conditions, sélectionnez l'opérateur logique **And** dans la liste **Logical operator used to combine conditions**, pour accepter uniquement les enregistrement répondant aux deux conditions.

8. Dans la vue **Basic settings** des composants **tLogRow**, sélectionnez l'option **Table (print values in cells of a table)** dans la zone **Mode**.

Exécuter le Job

- Sauvegardez votre Job puis appuyez sur **F6** pour l'exécuter.

```
[statistics] connected
.--------+------+---+--------.
|          accepted           |
|=-------+------+---+-------=|
|LastName|Gender|Age|City    |
|=-------+------+---+-------=|
|Adams   |M     |40 |Albany  |
|Harrison|M     |40 |New York|
|Arthur  |M     |20 |Albany  |
|Pierce  |M     |18 |New York|
|McKinley|M     |70 |Boston  |
|Monroe  |M     |60 |Chicago |
'--------+------+---+--------'
.---------+------+---+--------+-----------------------------------------------------------------.
|                                          rejected                                             |
|=--------+------+---+--------+-----------------------------------------------------------------=|
|LastName |Gender|Age|City    |errorMessage                                                     |
|=--------+------+---+--------+-----------------------------------------------------------------=|
|Van Buren|M     |73 |Chicago | LastName.length() < 9 failed                                    |
|Jefferson|F     |66 |New York| LastName.length() < 9 failed|Gender.compareTo("M") == 0 failed|
|Adams    |M     |9  |Albany  |Age.compareTo(10) > 0 failed                                     |
|Jefferson|M     |30 |Chicago | LastName.length() < 9 failed                                    |
|Carter   |F     |26 |Chicago |Gender.compareTo("M") == 0 failed                                |
|Roosevelt|F     |15 |Chicago | LastName.length() < 9 failed|Gender.compareTo("M") == 0 failed|
|Monroe   |M     |8  |Boston  |Age.compareTo(10) > 0 failed                                     |
|Quincy   |F     |83 |Albany  |Gender.compareTo("M") == 0 failed|Age.compareTo(80) < 0 failed   |
|Coolidge |M     |4  |Chicago |Age.compareTo(10) > 0 failed                                     |
'---------+------+---+--------+-----------------------------------------------------------------'
[statistics] disconnected
```

La première table liste les enregistrements des personnes de sexe masculin, âgées de 10 à 80 ans, dont le nom de famille comprend moins de 9 caractères. La seconde affiche tous les enregistrements ne correspondant pas aux conditions de filtre. A chaque enregistrement rejeté correspond un message d'erreur expliquant la raison du rejet.

Scénario 2 : Filtrer une liste de noms via différentes opérations logiques

Ce scénario se base sur le scénario précédent et filtre les données d'entrée afin que seuls les enregistrements des personnes de New York et Chicago soient acceptées. Sans modifier les paramètres de filtre du précédent scénario, des conditions avancées sont ajoutées pour permettre les opérations logiques OR et AND dans le même composant **tFilterRow**.

1. Double-cliquez sur le composant **tFilterRow** pour afficher sa vue **Basic settings**.

2. Cochez la case **Use advanced mode** et saisissez l'expression suivante dans le champ **Advanced** :

```
input_row.City.equals("Chicago") || input_row.City.equals("New York")
```

Cette expression définit deux conditions sur la colonne *City* des données d'entrée afin de filtrer les enregistrements contenant les villes de Chicago et New York, et utilise l'opérateur logique OR pour accepter les enregistrements satisfaisant l'une des deux conditions.

3. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job et appuyez sur la touche **F6** pour l'exécuter.

```
[statistics] connected
.--------+------+---+--------.
|          accepted          |
|=-------+------+---+-------=|
|LastName|Gender|Age|City    |
|=-------+------+---+-------=|
|Harrison|M     |40 |New York|
|Pierce  |M     |18 |New York|
|Monroe  |M     |60 |Chicago |
'--------+------+---+--------'
.---------+------+---+--------+-----------------------------------------------------------------------------------------.
|                                                   rejected                                                            |
|=--------+------+---+--------+-----------------------------------------------------------------------------------------=|
|LastName |Gender|Age|City    |errorMessage                                                                             |
|=--------+------+---+--------+-----------------------------------------------------------------------------------------=|
|Van Buren|M     |73 |Chicago | LastName.length() < 9 failed                                                            |
|Adams    |M     |40 |Albany  |advanced condition failed                                                                |
|Jefferson|F     |66 |New York| LastName.length() < 9 failed|Gender.compareTo("M") == 0 failed                          |
|Adams    |M     |9  |Albany  |Age.compareTo(10) > 0 failed|advanced condition failed                                  |
|Jefferson|M     |30 |Chicago | LastName.length() < 9 failed                                                            |
|Carter   |F     |26 |Chicago |Gender.compareTo("M") == 0 failed                                                        |
|Roosevelt|F     |15 |Chicago | LastName.length() < 9 failed|Gender.compareTo("M") == 0 failed                          |
|Monroe   |M     |8  |Boston  |Age.compareTo(10) > 0 failed|advanced condition failed                                  |
|Arthur   |M     |20 |Albany  |advanced condition failed                                                                |
|Quincy   |F     |83 |Albany  |Gender.compareTo("M") == 0 failed|Age.compareTo(80) < 0 failed|advanced condition failed|
|McKinley |M     |70 |Boston  |advanced condition failed                                                                |
|Coolidge |M     |4  |Chicago |Age.compareTo(10) > 0 failed                                                             |
'---------+------+---+--------+-----------------------------------------------------------------------------------------'
[statistics] disconnected
```

La liste des résultats du scénario précédent a été plus filtrée. Seuls les enregistrements contenant les villes de New York et Chicago sont acceptés.

tJoin

Propriétés du tJoin

Famille de composant	Processing	
Fonction	Le **tJoin** joint deux tables, en appliquant la correspondance exacte dans plusieurs colonnes. Il compare les colonnes du flux principal et les colonnes de référence du flux de référence (Lookup), et écrit les données du flux principal et/ou les données rejetées.	
Objectif	Il aide à vérifier la qualité des données de n'importe quelle source par rapport à une source de données de référence.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisable dans divers projets et Job Designs. Voir également le *Guide utilisateur du Studio Talend*.
	Include lookup columns in output	Cochez cette case pour inclure les colonnes de référence (lookup), que vous avez définies, dans le flux de sortie.
Key definition	*Input key attribute*	Sélectionnez la ou les colonne(s) du flux principal qui doivent être vérifiées par rapport à la colonne clé de référence (lookup).
	Lookup key attribute	Sélectionnez les colonnes clé de référence (lookup) que vous utiliserez comme référence, afin de les comparer aux colonnes du flux d'entrée.
	Inner join (with reject output)	Cochez cette case pour joindre d'abord les deux tables et ensuite rassembler les données rejetées du flux principal.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant n'est pas un composant de début (fond vert), il nécessite deux composants d'entrée, et un ou plusieurs composant(s) de sortie.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario 1 : Faire une correspondance exacte entre deux colonnes et écrire les données rejetées

Ce scénario décrit un Job à cinq composants dont le but est d'effectuer une correspondance exacte entre la colonne *firstnameClient* d'un fichier d'entrée, par rapport aux données du fichier de référence d'entrée, et la colonne *lastnameClient*, par rapport aux données du fichier d'entrée de référence. Les sorties de cette correspondance sont écrites dans des fichiers de sortie distincts : les données exactes sont écrites dans un fichier Excel, tandis que les données inexactes sont écrites dans un fichier délimité.

Dans ce scénario, vous avez déjà stocké les schémas d'entrée et les fichiers de référence dans le **Repository**. Pour plus d'informations à propos du stockage de métadonnées dans l'arborescence du **Repository**, consultez le *Guide utilisateur du Studio Talend*.

Déposer et relier les composants

1. Dans l'arborescence du **Repository**, développez le nœud **Metadata** ainsi que le nœud des fichiers (**File**) où vous avez stocké les schémas d'entrée, et glissez la métadonnée correspondante dans l'espace de modélisation graphique.

   La boîte de dialogue **[Components]** s'ouvre.

2. Sélectionnez **tFileInputDelimited** dans la liste et cliquez sur **OK** afin de fermer la boîte de dialogue.

Le composant **tFileInputDelimited** apparaît dans l'espace de modélisation. Le fichier d'entrée utilisé dans ce scénario est appelé *ClientSample*. Il contient quatre colonnes, y compris les deux colonnes *firstnameClient* et *lastnameClient* sur lesquelles vous voulez appliquer la correspondance exacte.

3. Répétez l'opération pour le deuxième fichier d'entrée que vous souhaitez utiliser comme référence, *ClientSample_Update* dans ce scénario.

4. Glissez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tJoin**, **tFileOutputExcel**, et **tFileOutputDelimited**.

5. Connectez les fichiers principal et de référence au **tJoin** à l'aide d'une connexion de type **Row > Main**. Le lien principal entre le fichier de référence et le **tJoin** s'affiche comme un lien *Lookup* (de référence) dans l'espace de modélisation.

6. Connectez le **tJoin** au **tFileOutputExcel** à l'aide d'une connexion **Main**, et le **tJoin** au **tFileOutputDelimited** avec une connexion **Inner join reject**.

Configurer les composants

1. Au besoin, double-cliquez sur les fichiers d'entrée principal et de référence afin d'afficher leur vue **Basic settings**. Toutes les propriétés sont automatiquement renseignées. Si vous ne définissez pas vos fichiers d'entrée dans le **Repository**, remplissez les champs manuellement, après avoir sélectionné **Built-in** dans le champ **Property Type**.

   Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.

2. Double-cliquez sur le **tJoin** afin d'afficher sa vue **Basic settings** et définir ses propriétés.

3. Cliquez sur le bouton situé à côté du champ **Edit schema** pour ouvrir une boîte de dialogue qui affiche la structure des données des fichiers d'entrée, et définissez les données que vous souhaitez passer aux composants de sortie, trois colonnes dans ce scénario, *idClient*, *firstnameClient* et *lastnameClient*.

4. Cliquez sur **OK** pour fermer la boîte de dialogue.

5. Dans la zone **Key definition** de l'onglet **Basic settings** du composant **tJoin**, cliquez sur le bouton **[+]** pour ajouter deux colonnes à la liste, puis sélectionnez les colonnes d'entrée et de sortie sur lesquelles la correspondance exacte sera appliquée, depuis les listes **Input key attribute** et **Lookup key attribute**, respectivement *firstnameClient* et *lastnameClient* dans cet exemple.

6. Cochez la case **Inner join (with reject output)** pour définir l'une des sorties comme table de rejet de l'inner join.

7. Double-cliquez sur le composant **tFileOutputExcel** pour afficher l'onglet **Basic settings** de sa vue **Component**, et définir ses propriétés.

8. Paramétrez le répertoire de destination ainsi que le nom de la feuille (**Sheet name**) et cochez la case **Include header**.

Sauvegarder et exécuter le Job

1. Double-cliquez sur le composant **tFileOutputDelimited** pour afficher l'onglet **Basic settings** de sa vue **Component**, et définir ses propriétés.

2. Paramétrez le répertoire de destination ainsi que les séparateurs de lignes (**Row separator**) et de champs (**Field separator**) et cochez la case **Include header**.

3. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

La sortie de la correspondance exacte des colonnes *firstnameClient* et *lastnameClient* est écrite dans le fichier Excel défini.

	A	B	C
1	idClient	firstnameClient	lastnameClient
2	28	Herbert	Eisenhower
3	36	Chester	Grant

Les données qui ne correspondent pas sont écrites dans le fichier délimité défini.

```
1  idClient;firstnameClient;lastnameClient
2  1;Dwight;Madison
3  2;Franklin;Jackson
4  3;Ronald;Buchanan
5  4;Bill;Cleveland
6  5;William;Harrison
7  6;William;Fillmore
8  7;Harry;Adams
9  8;Harry;McKinley
10 9;Herbert;Reagan
11 10;Lyndon;Jefferson
12 11;Bill;Jackson
13 12;John;Hayes
14 13;Ulysses;Reagan
```

tMap

Propriétés du tMap

Famille de composant	Processing	
Fonction	Le **tMap** est un composant avancé qui s'intègre au *Studio Talend* comme un plug-in.	
Objectif	Le **tMap** transforme et dirige les données à partir d'une ou plusieurs source(s) et vers une ou plusieurs destination(s).	
Basic settings	*Map editor*	Le Mapper est l'éditeur du **tMap**. Il vous permet de définir les propriétés d'aiguillage et de transformation des données. Si vous ne souhaitez pas gérer les erreurs d'exécution, vous pouvez utiliser l'option **Die on error** (cochée par défaut), en haut à droite de l'éditeur. Cette option arrête le Job en cas d'erreur.
	Mapping links display as	**Auto** : par défaut, les liens sont en forme de courbes. **Curves** : les liens du mapping sont en forme de courbes. **Lines** : les liens du mapping sont en forme de lignes droites. Cette dernière option améliore légèrement les performances.
	Temp data directory path	Saisissez le chemin du dossier dans lequel vous souhaitez stocker les données temporaires générées par le traitement des données de référence (lookup). Pour plus d'informations sur ce dossier, consultez le *Guide d'utilisation du Studio Talend*.
	Preview	L'aperçu est un instantané des données du Mapper. Il n'est visible que lorsque les propriétés du Mapper sont renseignées. La synchronisation de l'aperçu ne prend effet qu'après la sauvegarde des modifications.
	Use replicated join	Cochez cette case pour effectuer une jointure répliquée entre les flux d'entrée. En répliquant chaque table Lookup dans la mémoire, ce type de jointure ne requiert pas d'étape supplémentaire de shuffle-and-sort, ce qui accélère le processus. Notez que cette fonctionnalité est disponible uniquement en version Map/Reduce du composant. Vous devez vous assurer que la mémoire contient assez d'espace pour contenir toutes les tables Lookup.
Advanced settings	*Max buffer size (nb of rows)*	Saisissez la taille de la mémoire physique, en nombre de lignes, que vous souhaitez allouer aux données traitées.
	Ignore trailing zeros for BigDecimal	Cochez cette case pour ignorer les zéros non nécessaires dans le traitement de données BigDecimal.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant. Notez que cette case n'est pas disponible dans la version Map/Reduce de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Plusieurs utilisations sont possibles, de la simple réorganisation des champs de données aux transformations les plus complexes, telles que le multiplexage et le démultiplexage de données, la concaténation, l'inversion, le filtrage, etc.
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	L'utilisation du composant **tMap** requiert un niveau moyen de connaissances en Java afin d'exploiter au mieux ses fonctionnalités. Ce composant est utilisé comme une étape intermédiaire, c'est pourquoi il ne peut pas être utilisé comme composant de début ou de fin de Job.

Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.

Scénario 1 : Mapping simple de données avec filtre et jointure explicite

Le Job ci-dessous a pour objectif de lire des données d'un fichier .csv stocké dans le **Repository**, de rechercher des données d'un fichier de référence, dont le schéma est également stocké dans le **Repository** puis d'extraire des données de ces deux fichiers en fonction de filtres et d'envoyer ces données vers les fichiers de sortie et de rejet.

Construire le Job

1. Déposez deux composants **tFileInputDelimited**, un **tMap** et trois **tFileOutputDelimited** de la **Palette** dans l'espace de modélisation graphique.

2. Renommez respectivement les deux **tFileInputDelimited** *Cars* (voitures) et *Owners* (propriétaires).

3. Reliez les deux composants d'entrée au **tMap** à l'aide de liens **Row** > **Main** et nommez ces connexions *Cars_data* et *Owners_Data*, respectivement.

4. Reliez le **tMap** aux trois composants de sortie à l'aide de liens **Row** > **New Output (Main)** et nommez ces connexions, respectivement, *Insured*, *Reject_NoInsur* et *Reject_OwnerID*.

Configurer les composants

1. Double-cliquez sur le composant **tFileInputDelimited** nommé *Cars* pour paramétrer ses propriétés de base (**Basic settings**).

2. Sélectionnez **Repository** dans les champs **Property type** et **Schema**. Pour le schéma, sélectionnez la métadonnée *cars* dans la boîte de dialogue **[Repository Content]**. Le reste des champs est automatiquement renseigné lorsque vous sélectionnez la métadonnée appropriée dans la liste.

3. Double-cliquez sur le composant *Owners* et définissez à nouveau les propriétés. Sélectionnez la métadonnée appropriée, *owners* dans cet exemple.

   Dans ce scénario, les schémas d'entrée sont stockés dans le nœud **Metadata** de la vue **Repository** afin de les retrouver plus facilement. Pour plus d'informations concernant la création de métadonnées dans le Repository, consultez le *Guide utilisateur du Studio Talend*.

4. Double-cliquez sur le composant **tMap** pour ouvrir le Mapper. La zone Input (données en entrée) est déjà renseignée avec les schémas des composants d'entrée et la première table correspond au flux principal (**Main Row**). Les libellés de chacune des connexions d'entrée apparaissent en en-tête des tables.

5. Créez une jointure entre les deux tables d'entrée en faisant simplement glisser le champ *ID_Owner* de la table principale (*Cars_data*) vers le champ équivalent de la table de référence (*Owners_data*).

6. Définissez le lien en **Inner Join** en cliquant sur le bouton **tMap settings**, en cliquant dans le champ **Value** pour *Join Model*, en cliquant sur le bouton qui apparaît dans le champ et en sélectionnant **Inner Join** dans la liste **Options**.

7. Glissez et déposez le contenu de la table *Cars_data* vers la table *Insured*.

8. Glissez les colonnes *ID_Owner*, *Registration* et *ID_Reseller* de la table *Cars_data*, ainsi que la colonne *Name* de la table *Owners_data* dans la table *Reject_NoInsur*.

9. Glissez toutes les colonnes de la table *Cars_data* dans la table *Reject_OwnerID*.

   Pour plus d'informations concernant le mapping de données, consultez le *Guide utilisateur du Studio Talend*.

10. Cliquez sur le bouton **[+]** en haut de la table *Insured* afin d'ajouter une ligne de filtre.

    Glissez la colonne *ID_Insurance* de la table *Owners_data* vers la zone de filtre et saisissez la formule suivante d'exclusion des valeurs non définies : `Owners_data.ID_Insurance != null`.

    Avec ce filtre, la table *Insured* va rassemble tous les enregistrements qui possèdent un *ID Insurance*.

11. Cliquez sur le bouton **tMap settings** en haut de la table *Reject_NoInsur* et sélectionnez **true** pour **Catch lookup inner join reject** afin que de définir cette table comme le flux de rejet standard pour rassembler les enregistrements n'ayant pas d'*ID Insurance*.

12. Cliquez sur le bouton **tMap Settings** en haut de la table *Reject_OwnerID* puis sélectionnez **true** pour **Catch lookup inner join reject** afin que cette table rassemble les enregistrements du flux *Cars_data* ayant des IDs manquants ou ne correspondant pas.

Cliquez sur **OK** pour valider et revenir à l'espace de modélisation.

13. Double-cliquez sur chacun des composants de sortie afin de définir leurs propriétés. Si vous souhaitez créer un nouveau fichier, parcourez votre système jusqu'au dossier de destination et saisissez le nom du fichier avec son extension.

Cochez la case **Include header** pour réutiliser les libellés des colonnes du schéma d'entrée dans comme en-tête dans votre fichier de sortie.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Sauvegardez votre Job et appuyez sur **F6** pour exécuter le Job ou passez à l'onglet **Run Job** et cliquez sur le bouton **Run**.

   Les fichiers de sortie sont créés et ils contiennent les données définies.

Scénario 2 : Mapping de données avec rejet Inner join

Ce scénario, basé sur le scénario 1, requiert l'ajout d'un fichier d'entrée contenant les informations sur les revendeurs, ainsi que l'ajout de champs supplémentaires dans la table de sortie principale. De plus, deux filtres sont ajoutés sur les jointures Inner Join afin de récupérer les rejets spécifiques.

Construire le Job

1. Déposez un **tFileInputDelimited** et un **tFileOutputDelimited** dans l'espace de modélisation graphique et renommez les composants, respectivement *Resellers* et *No_Reseller_ID*.

2. Connectez le **tFileInputDelimited** au **tMap** à l'aide d'un lien **Row** > **Main** et renommez le lien *Resellers_data*.

3. Connectez le **tMap** au nouveau **tFileOutputDelimited** à l'aide d'un lien **Row** > **New Output**. Renommez le lien *Reject_ResellerID*.

Configurer les composants

1. Double-cliquez sur le composant *Resellers* pour définir ses propriétés dans la vue **Basic settings**.

2. Sélectionnez **Repository** dans la liste **Property Type** et sélectionnez le schéma *resellers*, dans la boîte de dialogue **[Repository Content]**. Les autres champs sont automatiquement renseignés.

   Dans ce scénario, les schémas d'entrée sont stockés sous le nœud **Metadata** du **Repository** afin de les retrouver facilement. Pour plus d'informations concernant la création de métadonnées dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

3. Double-cliquez sur le composant **tMap** pour ouvrir le **Map Editor**. Vous remarquerez que le schéma est automatiquement ajouté à la zone Input des schémas d'entrée.

4. Créez une jointure entre le flux d'entrée principal et le nouveau flux d'entrée en déposant la colonne *ID_Reseller* de la table *Cars_data* dans la colonne *ID_Reseller* de la table *Resellers_data*.

5. Cliquez sur le bouton **tMap settings** en haut de la table *Resellers_data* et sélectionnez **Inner Join** comme modèle de jointure (**Join Model**).

6. Glissez-déposez toutes les colonnes de la table *Resellers_data* sauf *ID_Reseller* dans la table de sortie principale, *Insured*.

Lorsque deux Inner Joins sont définis, vous devez soit définir deux tables de rejet inner join pour différencier les deux rejets, soit rassembler les deux rejets inner join dans une seule et même table de sortie, s'il n'y a qu'une seule sortie.

7. Dans la zone de sortie Output, cliquez sur le signe **[+]** pour ajouter une nouvelle table de sortie et nommez-la *Reject_ResellerID*.

8. Déposez toutes les colonnes de la table *Cars_data* dans la table *Reject_ResellerID*.

9. Cliquez sur le bouton **tMap Settings** puis sélectionnez true pour **Catch lookup inner join reject** afin de définir la dernière table comme étant la table du flux de sortie de rejets Inner Join. Si la jointure Inner Join ne peut être établie pour une entrée, les informations concernant les voitures correspondantes seront rassemblées dans le flux de sortie.

10. Appliquez maintenant un filtre sur chacune des sorties de rejet Inner Join afin de distinguer les deux types de rejets.

    Dans la première table de rejet (*Reject_OwnerID*), cliquez sur la flèche dotée d'un signe [+] pour ajouter une ligne de filtre et renseignez ce champ avec la formule suivante pour ne recueillir que les rejets liés au champ *OwnerID* non défini : `Owners_data.ID_Owner==null`

11. Dans la seconde table de rejet Inner Join (*Reject_ResellerID*), renouvelez l'opération et utilisez la formule suivante : `Resellers_data.ID_Reseller==null`

Cliquez sur **OK** pour enregistrer les changements et fermer le Mapper.

12. Double-cliquez sur le composant *No_Reseller_ID* pour afficher sa vue **Basic settings**.

Dans ce scénario, spécifiez simplement le chemin d'accès au fichier de sortie et cochez la case **Include Header**. Laissez les autres paramètres tels qu'ils sont.

13. Pour comprendre le fonctionnement du Mapper, dans ce scénario, retirez du fichier *Resellers.csv* les lignes correspondant aux *Reseller ID 5 et 8*.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Exécutez le Job via la touche **F6** du clavier, ou à partir de la vue **Run**.

   Les quatre fichiers de sortie sont tous créés dans le dossier défini dans les propriétés de la vue **Component**.

Remarquez que dans le fichier de rejet lié à l'Inner Join, *NoResellerID.csv*, le champ *ID_Owners* correspond aux entrées du fichier *Cars* dont le Reseller ID est égal à 5 et 8, manquant dans le fichier Resellers.csv.

Scénario 3 : Mapping de jointure en cascade

Un troisième scénario plus avancé et basé sur le scénario 2 requiert l'ajout d'une nouvelle entrée contenant les informations d'assurance, par exemple.

Dans le **Mapper**, mettez en place une jointure **Inner Join** entre deux tables de référence (Lookup) des flux secondaires *Owners* et *Insurance* afin de gérer les références en cascade et par conséquent de récupérer les informations de la table *Insurance* via les données de la table *Owners*.

Scénario 4 : Mapping avancé avec filtres, jointures explicites et rejet Inner join

Ce scénario présente un Job utilisant le **tMap** et permettant de trouver les clients propriétaires d'une voiture de marque définie et avoir entre 2 et 6 enfants (inclus) dans un but de vente incitative.

Construire le Job

1. Cliquez-déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : trois **tFileInputDelimited**, un **tMap** et deux **tFileOutputDelimited**.

2. Connectez les composants d'entrée au **tMap** à l'aide de connexions de type **Main row**.

   Faites attention au premier composant que vous connectez car son flux de données sera automatiquement un flux **Main** (principal). Et toutes les autres connexions seront alors de type **Lookup** (flux de référence). Dans ce scénario, le composant d'entrée *Owners* sera relié au **tMap** par le flux **Main**.

Configurer les composants

1. Définissez les propriétés de chaque composant d'entrée dans l'onglet **Basic settings**. Configurez les propriétés du composant *Owners*.

2. Sélectionnez **Repository** dans la liste **Property Type** et sélectionnez le schéma *resellers*, dans la boîte de dialogue **[Repository Content]**. Les autres champs sont automatiquement renseignés.

   Dans ce scénario, les schémas d'entrée sont stockés sous le nœud **Metadata** du **Repository** afin de les retrouver facilement. Pour plus d'informations concernant la création de métadonnées dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

   Répétez ces étapes pour les autres composants d'entrée : *Cars* et *Resellers*. Ces deux flux de référence (**Lookup**) renseigneront les tables secondaires (lookup) de la zone d'entrée du **Map Editor**.

3. Double-cliquez sur le composant **tMap** pour ouvrir le **Map Editor** et définir le mapping et les filtres.

   Configurez une jointure explicite entre le flux **Main** *Owner* et le flux **Lookup** *Cars* en glissant-déposant la colonne *ID_Owner* de la table *Owner* dans la colonne *ID_Owner* de la table *Cars*. La jointure explicite s'affiche avec une clé de hachage.

4. Dans le champ **Expr. Key** de la colonne *Make* (Marque), saisissez (en Java) le filtre à appliquer. Dans ce cas d'utilisation, saisissez simplement *"BMW"* puisque ce qui est recherché ce sont les propriétaires (*Owners*) de cette marque en particulier.

5. Effectuez une jointure en cascade entre les deux tables de référence *Cars* et *Resellers*, sur la colonne *ID_Reseller*, afin de retrouver les informations sur les propriétaires et plus précisément le nombre d'enfants qu'ils ont.

6. Comme vous souhaitez exclure les valeurs nulles dans une table séparée et les exclure de la sortie standard, cliquez sur le bouton **tMap Settings** puis sélectionnez **Inner Join** comme modèle de jointure pour chaque table **Lookup** à filtrer.

7. Dans les paramètres du **tMap**, vous pouvez sélectionner une des options suivantes : **Unique match**, **First match**, ou **All Matches**. Dans cet exemple, sélectionnez l'option **All matches**. Ainsi, si plusieurs correspondances sont trouvées dans l'Inner Join (i.e. les lignes répondant à la jointure explicite ainsi qu'au filtre), elles seront toutes prises en compte dans le flux de sortie (soit dans le flux de rejet, soit dans le flux de sortie standard).

   L'option **Unique match** fonctionne de la même manière que l'option **Last match**. Les options **First match** et **All matches** fonctionnent comme l'indiquent leurs noms (respectivement première correspondance et toutes les correspondances).

8. Puis dans la zone **Output** du **Map Editor**, ajoutez deux tables : une pour les correspondances répondant à la jointure explicite et au filtre et une pour les rejets.

9. Déposez toutes les colonnes de la table *Owners*, les colonnes *Registration*, *Make* et *Color* de la table *Cars* ainsi que les colonnes *ID_Reseller* et *Name_Reseller* de la table *Reseller* dans la table de sortie principale.

10. Déposez toutes les colonnes de la table *Owners* dans la table de rejets.

11. Cliquez sur le bouton **Filter** en haut de la table de sortie principale afin d'afficher la zone d'expression **Filter**.

    Saisissez une instruction afin de réduire le nombre de lignes chargées dans le flux principale de sortie. Dans cet exemple, saisissez : `Owners.Children_Nr >=2 && Owners.Children_Nr <= 6`.

12. Dans la table de rejets, cliquez sur le bouton **tMap settings** et définissez les types de rejets.

    Sélectionnez **true** pour **Catch output reject** afin de collecter des données concernant les détenteurs de voitures BMW ayant moins de deux enfants ou plus de six enfants.

    Sélectionnez **true** pour **Catch lookup inner join reject** afin de collecter des données concernant les propriétaires d'autres marques et ceux pour qui l'information *Reseller* est introuvable.

Cliquez sur **OK** pour valider le mapping et fermer l'éditeur.

Dans l'éditeur graphique, cliquez-droit sur le **tMap** et créez une connexion de sortie vers les composants correspondants.

13. Configurez les paramètres des composants de sortie dans l'onglet **Basic settings**.

    Dans ce scénario, spécifiez simplement les chemins d'accès aux fichiers de sortie et cochez la case **Include Header**. Laissez les autres paramètres tels qu'ils sont.

Exécuter le Job

1. Enregistrez votre Job.

2. Appuyez sur **F6** pour exécuter votre Job. Le fichier de sortie principal contient les informations concernant les détenteurs de BMW ayant entre deux et six enfants, et le fichier de rejet contient les informations concernant les autres possesseurs de voiture(s).

Scénario 5 : Mapping avancé avec filtres et différents rejets

Ce scénario est une version modifiée du scénario précédent. Il décrit un Job qui applique des filtres pour limiter la recherche aux possesseurs de voitures BMW ou Mercedes ayant entre deux et six enfants et divise les données sans correspondance en différents flux de rejets.

Construire le Job

1. Reprenez le Job de *Scénario 4 : Mapping avancé avec filtres, jointures explicites et rejet Inner join*.

2. Déposez un nouveau **tFileOutputDelimited** de la **Palette** dans l'espace de modélisation graphique et nommez-le *Rejects_BMW_Mercedes*.

3. Connectez le composant **tMap** au nouveau composant de sortie à l'aide d'une connexion **Row** et nommez cette connexion selon la fonction du composant de sortie.

   Ce lien deviendra le nom de la nouvelle table de sortie dans le **Map Editor**.

4. Renommez les connexions et les composants de sortie existants afin de présenter leur fonction.

   Les tables de sortie dans le **Map Editor** seront automatiquement renommées selon le libellé des liens. Dans cet exemple, renommez les connexions de sortie *BMW_Mercedes_withChildren* et *Owners_Other_Makes*, respectivement.

Configurer les composants

1. Double-cliquez sur le composant **tMap** afin de lancer le **Map Editor** pour modifier le mapping et les filtres.

   Remarquez que la zone de sortie contient une nouvelle table de sortie, vide, nommée *Rejects BMW_Mercedes*. Vous pouvez ajuster la position de la table en la sélectionnant et en cliquant sur les flèches pointant vers le haut ou vers le bas, en haut de la zone de sortie.

2. Supprimez le filtre *BMW* de la table *Cars* dans la zone d'entrée.

3. Cliquez sur le bouton **Filters** pour afficher le champ **Filter** et saisissez un nouveau filtre afin de limiter la recherche aux seules marques de voitures BMW et Mercedes. L'instruction est la suivante : `Cars.Make.equals("BMW") || Cars.Make.equals("Mercedes")`.

4. Sélectionnez toutes les colonnes de la table principale de sortie et déposez-les dans la nouvelle table de sortie.

   Sinon, vous pouvez glisser les colonnes des tables d'entrée correspondantes dans la nouvelle table de sortie.

5. Cliquez sur le bouton **tMap settings** en haut de la nouvelle table de sortie et sélectionnez **true** pour **Catch output reject** afin de collecter les données concernant les détenteurs de voiture(s) BMW ou Mercedes ayant moins de deux enfants ou plus de six.

6. Dans la table *Owners_Other_Makes*, sélectionnez **true** pour **Catch lookup inner join reject** afin de collecter les données concernant les possesseurs de voitures d'autres marques et ceux dont l'information *reseller* est introuvable.

7. Cliquez sur **OK** pour valider le mapping et fermer le **Map Editor**.

8. Configurez les propriétés des composants de sortie dans leur vue **Basic settings** respective. Dans ce scénario, définissez simplement les nouveaux chemins d'accès aux fichiers de sortie et cochez la case **Include Header**.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** pour l'exécuter.

   Le contenu du flux de sortie principal montre que les lignes filtrées sont bien passées.

Scénario 6 : Mapping avancé avec rechargement de la base de référence à chaque ligne

Le scénario suivant décrit un Job permettant de récupérer certaines informations sur des personnes dans une base de données de référence, en se basant sur une jointure effectuée sur leur âge. Le flux principal des données source est lu à partir d'une table MySQL appelée *people_age* contenant certains détails sur les personnes, comme leur identifiant (numérique), leur nom et prénom (alphanumérique) ainsi que leur âge (numérique), qui peut être soit 40, soit 60 ans. On a intentionnellement réduit le nombre d'entrées de cette table.

Les informations de référence sont également stockées dans une table MySQL appelée *large_data_volume*. Cette table de référence contient un certain nombre d'entrées, dont le nom des villes dans lesquelles les personnes enregistrées dans la table principale sont allées. Le nombre d'entrées a été restreint pour plus de clarté mais, dans le cas d'une utilisation normale, le Job ci-dessous a des résultats plus significatifs quand on traite un gros volume de données de référence.

Pour optimiser les performances, un composant de connexion aux bases de données est utilisé au début de ce Job pour ouvrir une connexion à la base de données de référence afin de ne pas ouvrir et refermer une connexion à chaque ligne de la table de référence chargée.

Un filtre d'expression (expression filter) est appliqué à ce flux de référence, qui permet de ne sélectionner que les données des personnes dont l'âge est 60 ou 40 ans. Ainsi, seules les lignes pertinentes de la table de référence sont enregistrées pour chaque ligne du flux principal.

Ce Job présente donc comment, à partir d'un nombre limité de lignes de flux principal, on peut optimiser la base de données de référence pour n'enregistrer que les résultats correspondant à la clé d'expression (expression key).

En général, comme le chargement des références s'effectue pour chacune des lignes de flux principal, il est plus intéressant d'utiliser cette option quand on ne traite qu'un nombre limité de lignes de flux principal et un grand nombre de lignes de référence.

La jointure s'effectue au niveau du champ *age*. Les informations de la base de référence sont ensuite chargées pour chacune des lignes du flux principal en entrée, en utilisant l'option de chargement appropriée dans l'éditeur du composant **tMap**.

Dans cet exemple, les métadonnées ont été préparées pour le composant de connexion et les composants source. Pour plus d'informations sur les métadonnées de connexion aux bases de données et les métadonnées de création de schémas des tables, consultez la section dédiée au sujet dans le *Guide utilisateur du Studio Talend*.

Le Job est formé de cinq composants, quatre composants de bases de données et un composant de mapping.

Construire le Job

1. A partir du nœud **Metadata** > **Db Connections** dans le **Repository**, glissez le schéma de base de données dans l'éditeur graphique. Dans cet exemple, la table source s'appelle *people_age*.

2. Lorsque vous déposez le composant, une liste apparaît à l'écran : sélectionnez le composant **tMysqlInput**.

3. Déposez le schéma de base de donnée de référence à partir du nœud **Metadata** > **Db Connections**. Dans cet exemple, la table de référence s'appelle *large_data_volume*.

4. De la même manière, glissez une connexion à la base de données du nœud **Db Connections** dans le **Repository** dans l'éditeur graphique. Dans la boîte de dialogue **[Components]**, sélectionnez le composant **tMysqlConnection**. Ce composant crée une connexion permanente à la base de données de référence afin d'éviter que la connexion ne se referme au chargement que chaque ligne de référence.

5. A partir de la **Palette**, déposez dans l'éditeur graphique le composant **tMap** (famille **Processing**) et les composants **tMysqlCommit** et **tMysqlOutput** (famille **Database**).

6. Reliez maintenant tous les composants entre eux : faites un clic-droit sur le **tMysqlInput** correspondant à la table *people* et maintenez le bouton enfoncé tout en déplaçant le curseur jusqu'au **tMap**.

7. Relâchez le bouton lorsque vous êtes sur le **tMap**, le flux principal de lignes est créé automatiquement.

8. Renommez le lien de type **Main** en *people*, pour pouvoir identifier plus facilement le flux de données principal.

9. Effectuez la même opération pour connecter la table de référence (*large_data_volume*) au **tMap** et le **tMap** au **tMysqlOutput**.

10. Une boîte de dialogue apparaît et vous devez donner un nom au lien de sortie. Dans cet exemple, le flux de sortie s'appelle *people_mixandmatch.*

11. Renommez également le lien de connexion à la base de référence en *large_volume* pour pouvoir identifier plus facilement le flux de données.

12. Connectez le **tMysqlConnection** au **tMysqlInput** via un lien de type **OnSubjobOk**.

13. Connectez le composant **tMysqlInput** au **tMysqlCommit** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

Configurer les composants

1. Double-cliquez ensuite sur le composant **tMap** pour ouvrir l'éditeur graphique de mapping.

2. La table **Output** (qui s'est créée automatiquement quand vous avez lié le **tMap** au **tMysqlOutput**) sera composée des lignes concordantes du flux de référence (*large_data_volume*) et du flux principal (*people_age*).

   Sélectionnez les lignes du flux principal que vous voulez passer en sortie et faites les glisser vers la table de sortie (dans le coin supérieur droit de l'éditeur de mapping) : ils y seront copiés.

   Dans cet exemple, la sélection des lignes du flux principal inclut les champs suivants : *id*, *first_name*, *last_Name* et *age*.

   Dans la table de référence, la colonne *city* est sélectionnée.

   Faites glisser les colonnes sélectionnées des tables d'entrée (*people* et *large_volume*) vers la table de sortie.

3. Créez maintenant la jointure entre le flux principal et le flux de référence.

   Sélectionnez la colonne *age* dans la table du flux principal (en haut) et faites-la glisser vers la colonne *age* de la table du flux de référence (*large_volume* dans cet exemple).

   L'icône d'une clé apparaît à côté de l'expression liée dans la table de référence. La jointure est maintenant établie.

4. Cliquez sur le bouton **tMap Settings** puis sur le bouton **[...]** correspondant à **Lookup Model** et sélectionnez l'option **Reload at each row** dans la boîte de dialogue **[Options]** pour que la base de référence soit chargée à nouveau pour chaque nouvelle ligne traitée.

5. De la même manière, paramétrez **Match Model** comme **All matches** dans la table de référence pour récupérer dans le flux de sortie toutes les occurrences de *age* qui correspondent.

6. Implémentez ensuite le filtre, établi à partir de la colonne *age*, dans la table de référence. Le champ **GlobalMapKey** est créé automatiquement lorsque vous sélectionnez l'option **Reload at each row**. Vous pouvez donc utiliser cette expression pour filtrer les données de référence de manière dynamique, pour que seules les informations pertinentes soient chargées avec le flux principal.

   Comme l'a souligné l'introduction de ce scénario, le flux de données principal contient uniquement les personnes dont l'âge est de 40 ou 60. Pour éviter de charger toutes les lignes de la table de référence, dont les âges sont différents de 40 et 60, vous pouvez utiliser la valeur âge du flux principal comme variable globale pour alimenter le filtre de référence.

7. Glissez la colonne *Age* de la table de flux principal vers le champ **Expr.** dans la table de référence.

8. Renseignez ensuite le nom de la variable dans le champ **globalMap Key**, en utilisant une expression. Dans cet exemple, l'expression est : `"people.Age"`. Cliquez sur le bouton **OK** pour enregistrer les paramètres de mapping et retourner dans l'éditeur graphique.

9. Pour terminer l'implémentation du filtrage dynamique du flux de référence, vous devez maintenant ajouter une clause WHERE à la requête de la base de données d'entrée.

10. A la fin du champ **Query**, juste après la commande `Select`, entrez la clause de type WHERE suivante :

```
WHERE AGE ='"+((Integer)globalMap.get("people.Age"))+"'".
```

11. Assurez-vous que le type correspond à la colonne utilisée comme variable. Dans cet exemple, la donnée *Age* est du type `Integer`. Veillez aussi à utiliser la variable de la manière dont vous l'avez paramétrée dans le champ **globalMap key** de l'éditeur de mapping.

12. Double-cliquez ensuite sur le composant **tMysqlOutput** et vérifiez que le schéma correspond aux paramètres de mapping.

13. Cochez la case **Use an existing connection** afin d'utiliser la connexion créée.

14. Spécifiez dans le champ **Table** le nom de la table cible.

    Dans les listes **Action on table** et **Action on data**, sélectionnez les actions que vous souhaitez effectuer.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Cliquez sur l'onglet **Run** au bas de l'espace de modélisation graphique, afin d'afficher l'onglet d'exécution.

3. Dans la vue **Debug Run**, cliquez sur le bouton **Traces Debug** pour voir la progression du traitement des données.

   Pour une meilleure visibilité, il est possible d'agrandir la vue du Job dans l'éditeur graphique lors de l'exécution en double-cliquant simplement sur l'onglet portant le nom du Job.

Les données de référence sont chargées à nouveau pour chaque ligne du flux principal qui correspond à la contrainte d'âge. Toutes les correspondances à la condition *age* sont récupérées à partir des lignes de la table de référence et regroupées dans le flux de sortie.

Si vous vérifiez les données contenues dans la table *people_mixandmatch* nouvellement créée, vous y trouverez toutes les copies basées sur *age* correspondant aux différentes personnes dont l'âge vaut 40 ou 60, ainsi que les villes où ils sont allés.

id	First_Name	Last_...	AGE	CITY
1	Rutherford	Polk	40	Montpelier
1	Rutherford	Polk	40	Austin
1	Rutherford	Polk	40	Providence
1	Rutherford	Polk	40	Lansing
1	Rutherford	Polk	40	Denver
1	Rutherford	Polk	40	Jefferson City
2	Harry	Tyler	40	Montpelier
2	Harry	Tyler	40	Austin
2	Harry	Tyler	40	Providence
2	Harry	Tyler	40	Lansing
2	Harry	Tyler	40	Denver
2	Harry	Tyler	40	Jefferson City
3	Martin	John...	60	Harrisburg
4	Rutherford	Taylor	40	Montpelier

Scénario 7 : Mapping avec des tables de sortie

Le scénario suivant décrit un Job traitant des flux de rejet sans les séparer du flux principal.

Construire le Job

1. Dans la vue **Repository**, cliquez sur **Metadata** > **File delimited**. Glissez-déposez la métadonnée *customers* dans l'espace de modélisation graphique.

   La métadonnée *customers* contient des informations relatives aux clients, comme leur ID, leur nom, leur adresse, etc.

   Pour plus d'informations concernant la centralisation des métadonnées, consultez le *Guide utilisateur du Studio Talend*.

2. Une boîte de dialogue s'ouvre et vous demande de choisir le type de composant que vous souhaitez utiliser. Sélectionnez le composant **tFileInputDelimited** et cliquez sur **OK**.

3. Ensuite, glissez-déposez la métadonnée *states*. Sélectionnez le même composant dans la boîte de dialogue et cliquez sur **OK**.

La métadonnée *states* contient l'identifiant de l'état, ainsi que son nom.

4. Glissez-déposez de la **Palette** dans l'espace de modélisation graphique un **tMap** et deux composants **tLogRow**.

5. Reliez le composant *customers* au **tMap** à l'aide d'un lien **Row** > **Main**.

6. Reliez le composant *states* au **tMap**, en utilisant un lien **Row** > **Main**. Il sera automatiquement défini comme **Lookup**.

Configurer les composants

1. Double-cliquez sur le composant **tMap** afin d'ouvrir le **Map Editor**.

   Glissez-déposez la colonne *idState* dans la première ligne de la seconde table, afin de créer une jointure.

   Cliquez sur le bouton **tMap settings** et sélectionnez **Inner Join** pour **Join Model**.

2. Cliquez sur le bouton **Property settings** en haut de la zone d'entrée afin d'ouvrir la boîte de dialogue **[Property settings]** et décochez la case **Die on error** si vous souhaitez gérer les erreurs d'exécution.

   La table **ErrorReject** est maintenant créée.

3. Sélectionnez les colonnes *id*, *idState*, *RegTime* et *RegisterTime* de la table d'entrée et déposez-les dans la table **ErrorReject**.

4. Cliquez sur le bouton **[+]** en haut à droite de l'éditeur afin d'ajouter une table de sortie. Dans la boîte de dialogue qui s'ouvre, sélectionnez **New output**. Dans le champ à côté, saisissez le nom de la table, *out1*. Cliquez sur **OK**.

5. Déposez les colonnes suivantes des tables d'entrée dans la table *out1* : *id*, *CustomerName*, *idState* et *LabelState*.

   Ajoutez deux colonnes *RegTime* et *RegisterTime* à la fin de la colonne *out1* et paramétrez leur format de date : `"dd/MM/yyyy HH:mm"` et `"yyyy-MM-dd HH:mm:ss.SSS"`, respectivement.

6. Cliquez sur le champ **Expression** pour *RegTime* et appuyez sur **Ctrl+Espace** pour afficher la liste d'autocomplétion. Sélectionnez `TalendDate.parseDate`. Changez le modèle en `("dd/MM/yyyy HH:mm",row1.RegTime)`.

7. Répétez l'opération pour la seconde colonne, mais changez le modèle en `("yyyy-MM-dd HH:mm:ss.SSS",row1.RegisterTime)`.

8. Cliquez sur le bouton **[+]** en haut à droite de l'éditeur, afin d'ajouter une table de sortie. Dans la boîte de dialogue qui s'ouvre, sélectionnez **Create join table from**, choisissez *Out1* et donnez le nom *rejectInner*. Cliquez sur **OK**.

9. Cliquez sur le bouton **tMap settings** et sélectionnez **true** pour **Catch lookup inner join reject**, afin de gérer les rejets.

10. Déposez les colonnes *id*, *CustomerName* et *idState* des tables d'entrée dans les colonnes correspondantes de la table *rejectInner*.

    Cliquez dans le champ **Expression** de la colonne *LabelState* et saisissez `"UNKNOWN"`.

11. Cliquez dans le champ **Expression** de la colonne *RegTime*, appuyez sur **Ctrl+Espace**, et sélectionnez `TalendDate.parseDate`. Changez le modèle en `("dd/MM/yyyy HH:mm",row1.RegTime.)`

12. Dans la colonne *RegisterTime*, appuyez sur **Ctrl+Espace**, et sélectionnez `TalendDate.parseDate`, mais changez le modèle en `("yyyy-MM-dd HH:mm:ss.SSS",row1.RegisterTime)`.

    Si les données de la ligne *row1* ont un modèle incorrect, elles seront retournées par le flux *ErrorReject*.

Cliquez sur **OK** afin de valider les modifications et fermer l'éditeur.

13. Double-cliquez sur le composant **tLogRow** afin d'afficher sa vue **Component**.

    Cliquez sur **Sync columns** pour récupérer la structure du schéma depuis le mapper si nécessaire.

    Dans la zone **Mode**, sélectionnez **Table**.

    Répétez l'opération avec le second **tLogRow**.

Exécuter le Job

1. Appuyez sur **Ctrl+S** afin de sauvegarder le Job.
2. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

   La console de la vue **Run** affiche les flux de sortie principal et le flux *ErrorReject*. Le flux de sortie principal unifie les données valides et les rejets Inner Join, alors que le flux *ErrorReject* contient les informations des erreurs concernant les lignes ayant un format de date non analysable.

```
+-------------------------------+-------+--------------+----------------+----------------------+
                                              tLogRow_1                                          |
+-------------------------------+-------+--------------+----------------+----------------------=|
|CustomerName                   |idState|LabelState    |RegTime         |RegisterTime          |
+-------------------------------+-------+--------------+----------------+----------------------=|
|Griffith Paving and Sealcoatin|7      |Connecticut   |03/11/2006 09:20|2001-01-17 06:26:40.000|
|Bill's Dive Shop               |35     |Ohio          |19/11/2004 15:48|2002-06-07 09:40:00.000|
|Childress Child Day Care       |60     |UNKNOWN       |16/02/2005 08:27|1990-04-01 21:00:00.000|
|Facelift Kitchen and Bath      |0      |UNKNOWN       |22/08/2002 09:55|1972-04-23 18:00:00.000|
|Terrinni & Son Auto and Truck  |5      |California    |28/06/2001 09:15|1982-04-19 10:26:40.000|
|Kermit the Pet Shop            |28     |Nevada        |17/08/2003 10:07|2006-05-27 17:00:00.000|
|Muh's Furniture Store          |15     |Iowa          |27/08/2000 03:13|1970-03-27 23:08:16.000|
```

tNormalize

Propriétés du tNormalize

Famille de composant	Processing/Fields	
Fonction	Le composant **tNormalize** normalise un flux entrant en fonction du standard SQL.	
Objectif	Le **tNormalize** permet d'améliorer la qualité de données et donc facilite la mise à jour des données.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant uniquement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Column to normalize	Sélectionnez la colonne du flux entrant sur laquelle est basée la normalisation.
	Item separator	Renseignez le séparateur délimitant les données du flux entrant. Le séparateur d'éléments se base sur des expressions régulières. Par conséquent, le caractère "." (caractère spécial dans les expressions régulières) doit être évité ou utilisé avec prudence.
Advanced settings	*Get rid of duplicated rows from output*	Cochez cette case pour supprimer les lignes doublons du flux de sortie. Cette fonctionnalité n'est pas disponible pour la version Map/Reduce de ce composant.
	Use CSV parameters	Cochez cette case pour prendre en compte les paramètres spécifiques aux fichiers CSV, notamment la manière de protéger les caractères dans le champ **escape mode** et le type de guillemet dans le champ **enclosure**.
	Discard the trailing empty strings	Cochez cette case pour ignorer les chaînes de caractères vides à la fin d'un groupe d'éléments.
	Trim resulting values	Cochez cette case pour supprimer les espaces en début et en fin de champ des données de résultat.

		Lorsque les cases **Discard the trailing empty string** et **Trim resulting values** sont cochées, le traitement de la première option s'effectue en premier.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n'est pas disponible dans la version Map/Reduce de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé comme composant intermédiaire dans un flux de données.	
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Pour un exemple scénario de Job Map/Reduce utilisant ce composant, consultez *Scénario 2 : Normaliser des données à l'aide de composants Map/Reduce*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario 1 : Normaliser des données

Ce scénario illustre un Job qui normalise une liste de mots-clés des sujets d'un forum puis affiche les résultats dans un table dans la console de la vue **Run**.

Cette liste n'est pas vraiment organisée et contient des espaces vides en début en fin de champs, ainsi que des mots-clés répétés.

```
ldap,
  db2, jdbc driver,
grid computing,  talend architecture  ,
```

```
content, environment,,
tmap,,
eclipse,
database,java,postgresql,
tmap,
database,java,sybase,
deployment,,
repository,
database,informix,java
```

Construire le Job

1. Cliquez et déposez les composants suivants dans l'espace de modélisation : **tFileInputDelimited**, **tNormalize**, **tLogRow**.

2. Reliez les composants à l'aide de liens **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le composant **tFileInputDelimited**, pour ouvrir sa vue **Basic settings**.

2. Dans le champ **File name**, spécifiez le chemin d'accès au fichier d'entrée à normaliser.

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir la boîte de dialogue **[Schema]** et configurez le schéma d'entrée en ajoutant une colonne nommée *Tags*. Cela fait, cliquez sur **OK** pour valider votre schéma et fermer la boîte de dialogue. Laissez les autres paramètres tels qu'ils sont.

4. Double-cliquez sur le composant **tNormalize** pour ouvrir sa vue **Basic settings**.

5. Vérifiez le schéma et, si nécessaire, cliquez sur le bouton **Sync columns** afin de synchroniser le schéma et le schéma du composant d'entrée.

6. Définissez la colonne sur laquelle est basée la normalisation.

   Dans ce scénario, le schéma d'entrée possède une seule colonne, *Tags*. Acceptez donc les paramètres par défaut.

7. Dans la vue **Advanced settings**, cochez les cases **Get rid of duplicate rows from output**, **Discard the trailing empty strings** et **Trim resulting values**.

8. Dans les propriétés du composant **tLogRow**, sélectionnez le mode **Table (print values in the cells of table)**.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Cliquez sur le bouton **Run**, dans la vue **Run** ou appuyez sur **F6** pour exécuter votre Job.

La liste est nettoyée, les doublons, et les espaces vides sont supprimés. Le résultat est affiché dans un tableau dans la console.

Scénario 2 : Normaliser des données à l'aide de composants Map/Reduce

Vous pouvez créer la version Map/Reduce du Job décrit précédemment à l'aide des composants Map/Reduce. Ce Job **Talend** Map/Reduce génère du code Map/Reduce et s'exécute nativement dans Hadoop.

Notez que les composants **Talend** Map/Reduce ne sont disponibles que si vous avez souscrit à l'une des solutions Big Data de **Talend** et que ce scénario ne peut être reproduit qu'avec des composants Map/Reduce.

Les données d'exemple utilisées dans ce scénario sont les mêmes que celles utilisées dans le Job décrit précédemment.

```
ldap,
  db2, jdbc driver,
grid computing,  talend architecture  ,
content, environment,,
tmap,,
eclipse,
database,java,postgresql,
tmap,
database,java,sybase,
deployment,,
repository,
database,informix,java
```

Etant donné que le studio **Talend** vous permet de convertir un Job **Map/Reduce** en Job **Standard** (non Map/Reduce), et vice-versa, vous pouvez convertir le scénario présenté précédemment afin de créer ce Job Map/Reduce. Ainsi, la plupart des composants utilisés peuvent garder leurs paramètres d'origine afin de réduire votre charge de travail pour la création de ce Job.

Avant de commencer ce scénario, assurez-vous de disposer des droits d'accès Hadoop nécessaires pour accéder à la distribution Hadoop utilisée. Procédez ensuite comme suit :

Convertir le Job

1. Dans le **Repository** de la perspective **Integration** du *Studio Talend*, cliquez-droit sur le Job que vous avez créé dans le scénario précédent afin d'ouvrir son menu contextuel. Sélectionnez ensuite **Edit properties**.

   L'assistant **[Edit properties]** s'ouvre alors. Notez que le Job doit être fermé avant que vous ne puissiez effectuer des changements dans cet assistant.

   L'assistant se présente comme dans la capture d'écran ci-dessous :

Notez que vous pouvez changer le nom du Job ainsi que les autres informations de description du Job à partir de cet assistant.

2. Cliquez sur **Convert to Map/Reduce Job**. Un Job Map/Reduce portant le même nom apparaît sous le sous-nœud **Map/Reduce Jobs** du nœud **Job Design**.

Si vous devez partir d'un Job Map/Reduce Job vierge, cliquez-droit sur le nœud **Job Design** ou sur le sous-nœud **Map/Reduce Jobs** et sélectionnez **Create Map/Reduce Job** dans le menu contextuel. Un Job vierge s'ouvre dans l'espace de modélisation graphique. Pour plus d'informations, consultez la section décrivant comment créer un Job Map/Reduce dans le *Guide de prise en main de Talend Open Studio for Big Data*.

Réorganiser les composants

1. Double-cliquez sur le Job Map/Reduce que vous venez de créer afin de l'ouvrir dans l'espace de modélisation graphique. La **Palette** de composants Map/Reduce s'ouvre et des composants peuvent être barrés car il n'ont pas de version Map/Reduce.

2. Cliquez-droit sur chacun des composants en question et sélectionnez **Delete** afin de les supprimer.

3. Déposez un composant **tHDFSInput** et un **tHDFSOutput** dans l'espace de modélisation graphique. Le **tHDFSInput** lit les données à partir de la distribution Hadoop utilisée et le **tHDFSOutput**, remplaçant le **tLogRow**, écrit les données dans cette distribution.

   Si vous partez d'un Job vierge, déposez également un **tNormalize**.

4. Connectez le **tHDFSInput** au **tNormalize** à l'aide d'un lien **Row > Main** et acceptez le schéma provenant du **tNormalize**.

5. Connectez également le **tNormalize** au **tHDFSOutput** à l'aide d'un lien **Row > Main**.

Configurer une connexion Hadoop

1. Cliquez sur **Run** afin d'ouvrir la vue correspondante et cliquez sur l'onglet **Hadoop Configuration** afin d'afficher cette vue et configurer la connexion Hadoop pour ce Job.

   La vue se présente comme dans la capture d'écran ci-dessous :

2. Dans la liste **Property type**, sélectionnez **Built-in**. Si vous avez créé une connexion afin qu'elle soit utilisable depuis le **Repository**, sélectionnez alors **Repository** afin que le *Studio Talend* réutilise les informations de connexion pour ce Job.

   Pour plus d'informations concernant la création d'une connexion Hadoop dans le **Repository**, consultez le chapitre décrivant le nœud **Hadoop cluster** du *Guide de prise en main de Talend Open Studio for Big Data*.

3. Dans la zone **Version**, sélectionnez la distribution Hadoop à utiliser et sa version. Si vous ne trouvez pas la distribution correspondant à la votre dans la liste, sélectionnez **Custom** afin de vous connecter à une distribution Hadoop non officiellement supportée par le *Studio Talend*.

   Pour un exemple d'utilisation étape par étape de cette option **Custom**, consultez *Connexion à une distribution Hadoop personnalisée*.

   En plus de l'évolution de Hadoop, veuillez noter les changements suivants :

- Si vous utilisez **Hortonworks Data Platform V2.2**, les fichiers de configuration de votre cluster peuvent utiliser des variables d'environnement comme *${hdp.version}*. Si vous êtes dans cette situation, vous devez configurer la propriété *mapreduce.application.framework.path* dans la table **Hadoop properties**, avec la valeur du chemin d'accès pointant explicitement vers l'archive du framework MapReduce de votre cluster. Par exemple :

```
mapreduce.application.framework.path=/hdp/apps/2.2.0.0-2041/mapreduce/
mapreduce.tar.gz#mr-framework
```

- Si vous utilisez **Hortonworks Data Platform V2.0.0**, le type de système d'exploitation utilisé pour la distribution Hadoop et pour le Job **Talend** doit être identique, par exemple Windows ou Linux. Sinon, vous devez utiliser le Jobserver de **Talend** afin d'exécuter un Job dans le même type de système d'exploitation que celui dans lequel la distribution **Hortonworks Data Platform V2.0.0** utilisée est exécutée. Pour plus d'informations concernant le Jobserver de **Talend**, consultez le *Guide d'installation et de migration Talend*.

4. Dans le champ **Name node**, saisissez l'emplacement du nœud maître, le NameNode, de la distribution utilisée. Par exemple, *hdfs://tal-qa113.talend.lan:8020*.

   Si vous utilisez une distribution MapR, vous pouvez simplement laisser *maprfs:///* dans ce champ. Le client MapR se charge de créer la connexion au moment de l'exécution. Notez que le client doit être correctement installé. Pour plus d'informations concernant la configuration d'un client MapR, consultez la page suivante sur le site de la documentation de MapR : http://doc.mapr.com/display/MapR/Setting+Up+the+Client (en anglais).

5. Dans le champ **Job tracker**, saisissez l'emplacement du JobTracker de votre distribution. Par exemple, *tal-qa114.talend.lan:8050*.

   Notez que le mot "Job" dans le terme "JobTracker" désigne les Jobs MR ou Map/Reduce décrits dans la documentation Apache disponible sur le site Internet d'Apache http://hadoop.apache.org/ (en anglais).

   Si vous utilisez YARN dans votre cluster Hadoop, comme **Hortonworks Data Platform V2.0.0** ou **Cloudera CDH4.3 + (mode YARN)**, vous devez spécifier l'emplacement du **Resource Manager** et non celui du Jobtracker. Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser :

   - Cochez la case **Set resourcemanager scheduler address** et saisissez l'adresse de l'ordonnanceur (Scheduler) dans le champ qui apparaît.
   - Cochez la case **Set jobhistory address** et saisissez l'emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory.
   - Cochez la case **Set staging directory** et saisissez le chemin d'accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l'exécution de programmes. Ce répertoire se trouve sous la propriété *yarn.app.mapreduce.am.staging-dir* dans les fichiers de configuration, notamment les fichiers *yarn-site.xml* et *mapred-site.xml* de votre distribution.
   - Cochez la case **Use datanode hostname** pour permettre au Job d'accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété *dfs.client.use.datanode.hostname* à *true*. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.

6. Si la distribution utilisée nécessite une authentification Kerberos, cochez la case **Use Kerberos authentication** et saisissez les détails d'authentification. Sinon, laissez cette case décochée.

   Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d'utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

   De plus, comme ce composant effectue des calculs Map/Reduce, vous devez également authentifier les services associés, comme le serveur de l'historique des Jobs et le gestionnaire de ressources ou le JobTracker,

selon votre distribution, dans le champ correspondant. Ces principaux se trouvent dans les fichiers de configuration de votre distribution. Par exemple, dans une distribution CDH4, le principal du gestionnaire de ressource est configuré dans le fichier *yarn-site.xml* et le principal de l'historique des Job dans le fichier *mapred-site.xml*.

Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case **Use a keytab to authenticate**. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**.

Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.

7. Dans le champ **User name**, saisissez votre nom d'utilisateur pour votre distribution. Si vous laissez ce champ vierge, le nom d'utilisateur de la machine hébergeant le *Studio Talend* est utilisé.

8. Dans le champ **Temp folder**, saisissez le chemin d'accès au dossier HDFS dans lequel sont stockés les fichiers temporaires générés pendant les opérations de Map/Reduce.

9. Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d'autres termes, changez le séparateur si celui-ci n'est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.

10. Laissez la case **Clear temporary folder** (Effacer le contenu du dossier temporaire) cochée, sauf si vous souhaitez garder ces fichiers temporaires.

11. Laissez la case **Compress intermediate map output to reduce network traffic** cochée, afin de réduire le temps de transfert des partitions du Mapper aux différents Reducers.

    Cependant, si le transfert des données dans le Job est négligeable, il est recommandé de décocher cette case afin de désactiver l'étape de compression, car la compression consomme des ressources supplémentaires du CPU.

12. Si vous souhaitez utiliser des propriétés Hadoop personnalisées, renseignez la table **Hadoop properties** avec la ou les propriété(s) à personnaliser. Lors de l'exécution, ces modifications écrasent les propriétés correspondantes par défaut utilisées par le Studio pour son moteur Hadoop.

    Pour plus d'informations concernant les propriétés requises par Hadoop, consultez la documentation Apache Hadoop sur le site http://hadoop.apache.org (en anglais), ou la documentation de la distribution Hadoop que vous utilisez.

13. Si le système Hadoop utilisé est Hortonworks Data Platform V1.2 ou Hortonworks Data Platform V1.3, vous devez définir des allocations de mémoire suffisantes pour que les opérations de map et reduce soient effectuées par le système Hadoop.

    Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations *map* et *reduce* dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux *1000*, ce qui est normalement adapté pour l'exécution de ces opérations.

    Si la distribution utilisée est YARN, les paramètres de mémoire à définir sont alors **Map (in Mb)**, **Reduce (in Mb)** et **ApplicationMaster (in Mb)**, respectivement. Ces champs permettent d'allouer dynamiquement de la mémoire aux opérations map et reduce et à l'ApplicationMaster de YARN.

Pour plus d'informations sur l'onglet **Hadoop Configuration**, consultez la section décrivant comment configurer la connexion Hadoop pour un Job **Talend** Map/Reduce dans le *Guide de prise en main de Talend Open Studio for Big Data*.

Pour plus d'informations sur le Resource Manager, son ordonnanceur et l'ApplicationMaster, consultez la documentation de YARN sur le site http://hortonworks.com/blog/apache-hadoop-yarn-concepts-and-applications/ (en anglais).

Pour plus d'informations concernant la définition des paramètres de mémoire pour YARN et MapReduce, consultez la documentation de la distribution utilisée comme, par exemple, celle d'Hortonworks : http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.0.6.0/bk_installing_manually_book/content/rpm-chap1-11.html (en anglais).

Configurer les composants d'entrée et de sortie

Configurer tHDFSInput

1. Double-cliquez sur le **tHDFSInput** pour ouvrir sa vue **Component**.

2. Cliquez sur le bouton ... situé à côté du champ **Edit schema** afin de vérifier si le schéma reçu dans les étapes précédentes a été correctement défini.

Notez que si vous avez créé ce Job à partir d'un Job vierge, vous devez cliquer sur le bouton afin d'ajouter manuellement ces colonnes au schéma. Si le schéma a été défini dans le **Repository**, vous pouvez sélectionner l'option **Repository** dans la liste **Schema** de la vue **Basic settings** afin de le réutiliser. Pour plus d'informations concernant la définition d'un schéma dans le **Repository**, consultez le chapitre décrivant la

gestion des métadonnées dans le *Guide utilisateur du Studio Talend* ou le chapitre décrivant le nœud **Hadoop cluster** du **Repository** dans le *Guide de prise en main de Talend Open Studio for Big Data*.

3. Si vous effectuez des changements dans le schéma, cliquez sur **OK** afin de valider ces changements puis acceptez la propagation proposée par la boîte de dialogue qui s'affiche.

4. Dans le champ **Folder/File**, saisissez le chemin d'accès au fichier source que vous souhaitez que le Job lise. Vous pouvez également parcourir votre système de fichiers afin de définir ce chemin.

   Si le fichier ne se trouve pas dans le système HDFS utilisé, vous devez le placer dans le système HDFS. Pour cela, vous pouvez par exemple utiliser le **tFileInputDelimited** et le **tHDFSOutput** dans un Job **Standard**.

Vérifier les composants de transformation

- Double-cliquez sur le **tNormalize** afin d'ouvrir sa vue **Component**.

Ce composant garde la configuration définie dans le Job d'origine dans les onglets **Basic settings** et **Advanced settings**. Il normalise la colonne *Tags* du flux d'entrée.

Configurer le tHDFSOutput

1. Double-cliquez sur le **tHDFSOutput** afin d'ouvrir sa vue **Component**.

2. Comme expliqué précédemment pour la vérification du schéma du **tHDFSInput**, procédez de la même manière afin de vérifier le schéma du **tHDFSOutput**. S'il ne correspond pas au schéma du composant précédent, le **tNormalize**, cliquez sur **Sync columns** afin de récupérer le schéma du **tNormalize**.

3. Dans le champ **Folder**, saisissez le chemin d'accès au dossier dans lequel vous souhaitez écrire les données. Vous pouvez également parcourir votre système de fichiers afin de définir ce chemin.

4. Dans la liste **Action**, sélectionnez l'opération que vous souhaitez exécuter sur le dossier en question. Si le dossier existe déjà, sélectionnez **Overwrite** (écraser). Sinon, sélectionnez **Create** (créer).

Exécuter le Job

Appuyez sur la touche **F6** pour exécuter le Job.

Une fois le Job exécuté, vous pouvez visualiser les résultats dans la console Web de HDFS.

Contents of directory /user/ychen/mapreduceJob/result/normalize

Goto : /user/ychen/mapreduceJo go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
_SUCCESS	file	0 KB	3	64 MB	2013-05-27 16:55	rw-r--r--	ychen	ychen
_logs	dir				2013-05-27 16:55	rwxr-xr-x	ychen	ychen
part-00000	file	0.1 KB	3	64 MB	2013-05-27 16:55	rw-r--r--	ychen	ychen
part-00001	file	0.09 KB	3	64 MB	2013-05-27 16:55	rw-r--r--	ychen	ychen

Go back to DFS home

Local logs

Log directory

Hadoop, 2013.

HDFS:/user/ychen/mapre

talend-cdh4-datanode2:50075/browseBloc

Other bookmarks

Go back to dir listing

Advanced view/download options

```
ldap

db2
jdbc driver

grid computing
talend architecture

content
environment


tmap
```

Les données normalisées ont été écrites dans le dossier spécifié.

Si vous souhaitez plus d'informations concernant le Job et son exécution, utilisez la console Web fournie par la distribution Hadoop que vous utilisez.

tReplace

Propriétés du tReplace

Famille de composant	Processing	
Fonction	Le composant **tReplace** effectue un Rechercher/Remplacer dans les colonnes d'entrée spécifiées.	
Objectif	Ce composant permet de nettoyer tous les fichiers avant traitement.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Deux colonnes en lecture seule sont automatiquement ajoutées au schéma de sortie : *Value* et *Match*.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Simple Mode Search / Replace	Cliquez sur le bouton **[+]** pour ajouter autant de conditions que vous le souhaitez. Les conditions sont exécutées l'une après l'autre pour chaque ligne. **Input column** : Sélectionnez la colonne du schéma sur laquelle effectuer le Rechercher/Remplacer. **Search** : Saisissez la valeur à chercher dans la colonne d'entrée. **Replace with** : Saisissez la valeur de remplacement. **Whole word** : Cochez cette case si la valeur recherchée constitue un mot entier. **Case sensitive** : Cochez cette case pour tenir compte de la casse. Notez que vous ne pouvez pas utiliser d'expression régulière dans ces colonnes.
	Use advanced mode	Cochez cette case si vous souhaitez utiliser des expressions régulières.

Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n'est pas disponible dans la version Map/Reduce de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant n'est pas un composant de début puisqu'il requiert un flux de données. Il requiert aussi un composant de sortie.	
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Remplacement multiple et filtrage de colonnes

Le Job suivant recherche et remplace différentes coquilles et fautes dans un fichier .CSV puis filtre une colonne avant de générer en sortie un nouveau fichier .CSV contenant les données modifiées.

- Cliquez et déposez les composants suivants dans l'espace de modélisation graphique : **tFileInputDelimited**, **tReplace**, **tFilterColumn** et **tFileOutputDelimited**.
- Connectez les composants à l'aide de liens de type **Main Row** via un clic-droit sur chaque composant.

- Sélectionnez le composant **tFileInputDelimited** et paramétrez les propriétés du flux d'entrée.

- Le fichier est un simple fichier .csv stocké localement. Ainsi, dans le champ **Filepath**, renseignez l chemin d'accès au fichier. Dans le champ **Row Separator**, renseignez le séparateur de lignes, ici le retour chariot. Dans le champ **Field Separator**, renseignez le séparateur de champs, ici un point virgule. Dans cet exemple, l'en tête (**Header**) est le nom des colonnes, et il n'y a ni pied de page (**Footer**) ni de nombre limite de lignes à traiter (**Limit**).

- Le fichier contient des caractères tels que : `*t`, `$`, `.` ou encore `Nikson` que l'on veut changer en `Nixon`, et `streat`, que l'on veut changer en `Street`.

Street	FirstName	Name	Amount
streat	John	Kennedy	98.30$
streat	Richad	Nikson	78.23$
streat	Richard	Nikson	78.2$
streat	toto	Nikson	78.23$
streat	Richard	Nikson	78.23$
street	Georges *t	bush	99.99$

- Dans le champ **Schema**, sélectionnez aussi l'option **Built-in**. Le schéma est composé de quatre colonnes de différents types : chaînes de caractères (string) ou de nombres décimaux (float).

- Sélectionnez le composant **tReplace** pour paramétrer les propriétés des Rechercher/Remplacer.

InputColumn	Search	Replace with	Whole w...	Case Sen...	
Amount	"."	","			
Street	"streat"	"Street"			
Amount	"$"	"£"			
Name	"Nikson"	"Nixon"			
FirstName	"*t"	""			

- Le schéma peut être synchronisé à partir du flux d'entrée.

- Cochez la case **Simple mode** car les paramètres de recherche peuvent être facilement définis sans avoir recours aux expressions régulières.

- Cliquez sur le signe **[+]** pour ajouter des lignes aux tableaux des paramètres.

- En premier paramètre, sélectionnez *amount* dans la colonne **Input column**. Dans la colonne **Search**, cherchez tous les *points* et dans la colonne **Replace** indiquez la valeur de remplacement, ici la *virgule*. Les valeurs doivent être mises entre guillemets.
- En deuxième paramètre, sélectionnez *Street* dans la colonne **Input column**. Dans la colonne **Search**, cherchez la chaîne de caractères *streat*. Dans la colonne **Replace**, renseignez la valeur de remplacement, ici *Street*.
- En troisième paramètre, sélectionnez à nouveau *Amount* dans la colonne **Input column**, saisissez *$* dans la colonne **Search** et **£** dans la colonne **Replace**.
- En quatrième paramètre, sélectionnez *Name* dans la colonne **Input column**. Dans le champ **Search**, saisissez *Nikson* dans la colonne **Search** et *Nixon* dans la colonne **Replace**.
- En cinquième paramètre, sélectionnez *Firstname* dans la colonne **Input column**. Dans le champ **Search**, saisissez la chaîne de caractère **t*, et ne mettez rien entre guillemets.
- L'option **Advanced mode** n'est pas utilisée pour ce scénario.
- Sélectionnez le composant suivant : **tFilterColumn**.

- Le composant **tFilterColumn** contient un éditeur de schéma permettant de créer un schéma de sortie basé sur le nom des colonnes du schéma d'entrée. Dans cet exemple, ajoutez une nouvelle colonne *empty_field* et changez l'ordre des colonnes pour obtenir le schéma suivant : *empty_field*, *Firstname*, *Name*, *Street*, *Amount*.
- Cliquez sur **OK** pour valider.
- Double-cliquez sur le **tFileOutputDelimited** afin d'afficher sa vue **Component**.

- Dans le champ **Schema**, sélectionnez l'option **Built-in** et cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
- Enregistrez le Job et exécutez-le en appuyant sur **F6**.

	John	Kennedy	Street	98,30£
	Richad	Nixon	Street	78,23£
	Richard	Nixon	Street	78,2£
	toto	Nixon	Street	78,23£
	Richard	Nixon	Street	78,23£
	Georges	bush	street	99,99£

La première colonne est vide, les caractères parasites ont été supprimés des autres colonnes, et *Nikson* a été remplacé par *Nixon*. La colonne *Street* a été déplacée, les décimales sont dorénavant séparées par une virgule et le symbole monétaire a été changé.

Scénario 2 : Remplacer des valeurs et filtrer des colonnes à l'aide de composants Map/Reduce

Vous pouvez réutiliser la version Map/Reduce du Job décrit précédemment en utilisant des composants Map/Reduce. Ce Job Map/Reduce **Talend** génère du code Map/Reduce et s'exécute de manière native dans Hadoop.

Notez que les composants **Talend** Map/Reduce ne sont disponibles que si vous avez souscrit à l'une des solutions Big Data de **Talend** et que ce scénario ne peut être reproduit qu'avec des composants Map/Reduce.

Les données d'exemple utilisées dans ce scénario sont les mêmes que celles utilisées dans le Job décrit précédemment et se présentent comme suit :

```
streat;John;Kennedy;98.30$
streat;Richad;Nikson;78.23$
streat;Richard;Nikson;78.2$
streat;toto;Nikson;78.23$
streat;Richard;Nikson;78.23$
street;Georges *t;bush;99.99$
```

Etant donné que le studio **Talend** vous permet de convertir un Job **Map/Reduce** en Job **Standard** (non Map/Reduce), et vice-versa, vous pouvez convertir le scénario expliqué plus tôt afin de créer ce Job Map/Reduce. Ainsi, la plupart des composants utilisés peuvent garder leurs paramètres d'origine afin de réduire votre charge de travail pour la création de ce Job.

Avant de commencer ce scénario, assurez-vous de disposer des droits d'accès Hadoop nécessaires pour accéder à la distribution Hadoop utilisée. Procédez ensuite comme suit :

Convertir le Job

1. Dans le **Repository** de la perspective **Integration** du *Studio Talend*, cliquez-droit sur le Job que vous avez créé dans le scénario précédent afin d'ouvrir son menu contextuel. Sélectionnez ensuite **Edit properties**.

   L'assistant **[Edit properties]** s'ouvre alors. Notez que le Job doit être fermé avant que vous ne puissiez effectuer des changements dans cet assistant.

   L'assistant se présente comme dans la capture d'écran ci-dessous :

   Notez que vous pouvez changer le nom du Job ainsi que les autres informations de description du Job à partir de cet assistant.

2. Cliquez sur **Convert to Map/Reduce Job**. Un Job Map/Reduce portant le même nom apparaît sous le sous-nœud **Map/Reduce Jobs** du nœud **Job Design**.

Si vous devez partir d'un Job Map/Reduce Job vierge, cliquez-droit sur le nœud **Job Design** ou sur le sous-nœud **Map/Reduce Jobs** et sélectionnez **Create Map/Reduce Job** dans le menu contextuel. Un Job vierge s'ouvre dans l'espace de modélisation graphique. Pour plus d'informations, consultez la section décrivant comment créer un Job Map/Reduce dans le *Guide de prise en main de Talend Open Studio for Big Data*.

Réorganiser les composants

1. Double-cliquez sur le Job Map/Reduce que vous venez de créer afin de l'ouvrir dans l'espace de modélisation graphique. La **Palette** de composants Map/Reduce s'ouvre et des composants sont barrés car il n'ont pas de version Map/Reduce.

2. Cliquez-droit sur chacun des composants en question et sélectionnez **Delete** afin de les supprimer.

3. Déposez un composant **tHDFSInput** et un **tHDFSOutput** dans l'espace de modélisation graphique. Le **tHDFSInput** lit les données à partir de la distribution Hadoop utilisée et le **tHDFSOutput** écrit les données dans cette distribution.

   Si vous partez d'un Job vierge, déposez également un **tReplace** et un **tFilterColumns**.

4. Connectez le **tHDFSInput** au **tReplace** à l'aide d'un lien **Row > Main** et acceptez le schéma provenant du **tReplace**.

5. Connectez le **tFilterColumns** au **tHDFSOutput** à l'aide d'un lien **Row > Main**.

Configurer une connexion Hadoop

1. Cliquez sur **Run** afin d'ouvrir la vue correspondante et cliquez sur l'onglet **Hadoop Configuration** afin d'afficher cette vue et configurer la connexion Hadoop pour ce Job.

   La vue se présente comme dans la capture d'écran ci-dessous :

2. Dans la liste **Property type**, sélectionnez **Built-in**. Si vous avez créé une connexion afin qu'elle soit utilisable depuis le **Repository**, sélectionnez alors **Repository** afin que le *Studio Talend* réutilise les informations de connexion pour ce Job.

   Pour plus d'informations concernant la création d'une connexion Hadoop dans le **Repository**, consultez le chapitre décrivant le nœud **Hadoop cluster** du *Guide de prise en main de Talend Open Studio for Big Data*.

3. Dans la zone **Version**, sélectionnez la distribution Hadoop à utiliser et sa version. Si vous ne trouvez pas la distribution correspondant à la votre dans la liste, sélectionnez **Custom** afin de vous connecter à une distribution Hadoop non officiellement supportée par le *Studio Talend*.

   Pour un exemple d'utilisation étape par étape de cette option **Custom**, consultez *Connexion à une distribution Hadoop personnalisée*.

   En plus de l'évolution de Hadoop, veuillez noter les changements suivants :

   - Si vous utilisez **Hortonworks Data Platform V2.2**, les fichiers de configuration de votre cluster peuvent utiliser des variables d'environnement comme *${hdp.version}*. Si vous êtes dans cette situation, vous devez configurer la propriété *mapreduce.application.framework.path* dans la table **Hadoop properties**, avec la valeur du chemin d'accès pointant explicitement vers l'archive du framework MapReduce de votre cluster. Par exemple :

     ```
     mapreduce.application.framework.path=/hdp/apps/2.2.0.0-2041/mapreduce/
     mapreduce.tar.gz#mr-framework
     ```

   - Si vous utilisez **Hortonworks Data Platform V2.0.0**, le type de système d'exploitation utilisé pour la distribution Hadoop et pour le Job **Talend** doit être identique, par exemple Windows ou Linux. Sinon, vous devez utiliser le Jobserver de **Talend** afin d'exécuter un Job dans le même type de système d'exploitation que celui dans lequel la distribution **Hortonworks Data Platform V2.0.0** utilisée est exécutée. Pour plus d'informations concernant le Jobserver de **Talend**, consultez le *Guide d'installation et de migration Talend*.

4. Dans le champ **Name node**, saisissez l'emplacement du nœud maître, le NameNode, de la distribution utilisée. Par exemple, *hdfs://tal-qa113.talend.lan:8020*.

   Si vous utilisez une distribution MapR, vous pouvez simplement laisser *maprfs:///* dans ce champ. Le client MapR se charge de créer la connexion au moment de l'exécution. Notez que le client doit être correctement installé. Pour plus d'informations concernant la configuration d'un client MapR, consultez la page suivante sur le site de la documentation de MapR : http://doc.mapr.com/display/MapR/Setting+Up+the+Client (en anglais).

5. Dans le champ **Job tracker**, saisissez l'emplacement du JobTracker de votre distribution. Par exemple, *tal-qa114.talend.lan:8050*.

   Notez que le mot "Job" dans le terme "JobTracker" désigne les Jobs MR ou Map/Reduce décrits dans la documentation Apache disponible sur le site Internet d'Apache http://hadoop.apache.org/ (en anglais).

   Si vous utilisez YARN dans votre cluster Hadoop, comme **Hortonworks Data Platform V2.0.0** ou **Cloudera CDH4.3 + (mode YARN)**, vous devez spécifier l'emplacement du **Resource Manager** et non celui du Jobtracker. Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser :

   - Cochez la case **Set resourcemanager scheduler address** et saisissez l'adresse de l'ordonnanceur (Scheduler) dans le champ qui apparaît.

   - Cochez la case **Set jobhistory address** et saisissez l'emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory.

   - Cochez la case **Set staging directory** et saisissez le chemin d'accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l'exécution de programmes. Ce répertoire se trouve sous

la propriété *yarn.app.mapreduce.am.staging-dir* dans les fichiers de configuration, notamment les fichiers *yarn-site.xml* et *mapred-site.xml* de votre distribution.

- Cochez la case **Use datanode hostname** pour permettre au Job d'accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété *dfs.client.use.datanode.hostname* à *true*. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.

6. Si la distribution utilisée nécessite une authentification Kerberos, cochez la case **Use Kerberos authentication** et saisissez les détails d'authentification. Sinon, laissez cette case décochée.

   Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d'utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

   De plus, comme ce composant effectue des calculs Map/Reduce, vous devez également authentifier les services associés, comme le serveur de l'historique des Jobs et le gestionnaire de ressources ou le JobTracker, selon votre distribution, dans le champ correspondant. Ces principaux se trouvent dans les fichiers de configuration de votre distribution. Par exemple, dans une distribution CDH4, le principal du gestionnaire de ressource est configuré dans le fichier *yarn-site.xml* et le principal de l'historique des Job dans le fichier *mapred-site.xml*.

   Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case **Use a keytab to authenticate**. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**.

   Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.

7. Dans le champ **User name**, saisissez votre nom d'utilisateur pour votre distribution. Si vous laissez ce champ vierge, le nom d'utilisateur de la machine hébergeant le *Studio Talend* est utilisé.

8. Dans le champ **Temp folder**, saisissez le chemin d'accès au dossier HDFS dans lequel sont stockés les fichiers temporaires générés pendant les opérations de Map/Reduce.

9. Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d'autres termes, changez le séparateur si celui-ci n'est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.

10. Laissez la case **Clear temporary folder** (Effacer le contenu du dossier temporaire) cochée, sauf si vous souhaitez garder ces fichiers temporaires.

11. Laissez la case **Compress intermediate map output to reduce network traffic** cochée, afin de réduire le temps de transfert des partitions du Mapper aux différents Reducers.

    Cependant, si le transfert des données dans le Job est négligeable, il est recommandé de décocher cette case afin de désactiver l'étape de compression, car la compression consomme des ressources supplémentaires du CPU.

12. Si vous souhaitez utiliser des propriétés Hadoop personnalisées, renseignez la table **Hadoop properties** avec la ou les propriété(s) à personnaliser. Lors de l'exécution, ces modifications écrasent les propriétés correspondantes par défaut utilisées par le Studio pour son moteur Hadoop.

    Pour plus d'informations concernant les propriétés requises par Hadoop, consultez la documentation Apache Hadoop sur le site http://hadoop.apache.org (en anglais), ou la documentation de la distribution Hadoop que vous utilisez.

13. Si le système Hadoop utilisé est Hortonworks Data Platform V1.2 ou Hortonworks Data Platform V1.3, vous devez définir des allocations de mémoire suffisantes pour que les opérations de map et reduce soient effectuées par le système Hadoop.

    Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations *map* et *reduce* dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux *1000*, ce qui est normalement adapté pour l'exécution de ces opérations.

    Si la distribution utilisée est YARN, les paramètres de mémoire à définir sont alors **Map (in Mb)**, **Reduce (in Mb)** et **ApplicationMaster (in Mb)**, respectivement. Ces champs permettent d'allouer dynamiquement de la mémoire aux opérations map et reduce et à l'ApplicationMaster de YARN.

Pour plus d'informations sur l'onglet **Hadoop Configuration**, consultez la section décrivant comment configurer la connexion Hadoop pour un Job **Talend** Map/Reduce dans le *Guide de prise en main de Talend Open Studio for Big Data*.

Pour plus d'informations sur le Resource Manager, son ordonnanceur et l'ApplicationMaster, consultez la documentation de YARN sur le site http://hortonworks.com/blog/apache-hadoop-yarn-concepts-and-applications/ (en anglais).

Pour plus d'informations concernant la définition des paramètres de mémoire pour YARN et MapReduce, consultez la documentation de la distribution utilisée comme, par exemple, celle d'Hortonworks : http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.0.6.0/bk_installing_manually_book/content/rpm-chap1-11.html (en anglais).

Configurer les composants d'entrée et de sortie

Configurer le tHDFSInput

1. Double-cliquez sur le **tHDFSInput** pour ouvrir sa vue **Component**.

2. Cliquez sur le bouton ... situé à côté du champ **Edit schema** afin de vérifier si le schéma reçu dans les étapes précédentes a été correctement défini.

Notez que si vous avez créé ce Job à partir d'un Job vierge, vous devez cliquer sur le bouton afin d'ajouter manuellement ces colonnes au schéma. Si le schéma a été défini dans le **Repository**, vous pouvez sélectionner l'option **Repository** dans la liste **Schema** de la vue **Basic settings** afin de le réutiliser. Pour plus d'informations sur comment définir un schéma dans le **Repository**, consultez le chapitre décrivant la gestion des métadonnées dans le *Guide utilisateur du Studio Talend* ou le chapitre décrivant le nœud **Hadoop cluster** du **Repository** dans le *Guide de prise en main de Talend Open Studio for Big Data*.

3. Si vous effectuez des changements dans le schéma, cliquez sur **OK** afin de valider ces changements puis acceptez la propagation proposée par la boîte de dialogue qui s'affiche.

4. Dans le champ **Folder/File**, saisissez le chemin d'accès au fichier source que vous souhaitez que le Job lise. Vous pouvez également parcourir votre système de fichiers afin de définir ce chemin.

   Si le fichier ne se trouve pas dans le système HDFS utilisé, vous devez le placer dans le système HDFS. Pour cela, vous pouvez par exemple utiliser le **tFileInputDelimited** et le **tHDFSOutput** dans un Job **Standard**.

Vérifier les composants de transformation

1. Double-cliquez sur le **tReplace** afin d'ouvrir sa vue **Component**.

Ce composant garde la configuration utilisée dans le Job d'origine. Il cherche des données d'entrée et les remplace celles spécifiées dans la colonne **Search** par les valeurs données dans la colonne **Replace with**.

2. Double-cliquez sur le **tFilterColumns** pour ouvrir sa vue **Component**.

Le composant garde le schéma utilisé dans le Job d'origine. Cependant, l'ordre des colonnes ne reste pas tel qu'après la réorganisation effectuée dans le Job précédent. Il reprend donc l'ordre d'origine.

Configurer le tHDFSOutput

1. Double-cliquez sur le **tHDFSOutput** pour ouvrir sa vue **Component**.

2. Comme expliqué précédemment pour la vérification du schéma du **tHDFSInput**, procédez de la même manière afin de vérifier le schéma du **tHDFSOutput**. S'il ne correspond pas au schéma du composant précédent, le **tFilterColumns**, cliquez sur **Sync columns** afin de récupérer le schéma du **tFilterColumns**.

3. Dans le champ **Folder**, saisissez le chemin d'accès au dossier dans lequel vous souhaitez écrire les données uniques. Vous pouvez également parcourir votre système de fichiers afin de définir ce chemin.

4. Dans la liste **Action**, sélectionnez l'opération que vous souhaitez exécuter sur le dossier en question. Si le dossier existe déjà, sélectionnez **Overwrite**. Sinon, sélectionnez **Create**.

Exécuter le Job

Appuyez sur la touche **F6** pour exécuter le Job.

Une fois le Job exécuté, vous pouvez visualiser les résultats dans la console Web de HDFS.

Si vous souhaitez plus d'informations concernant le Job et son exécution, utilisez la console Web fournie par la distribution Hadoop que vous utilisez.

tSampleRow

Propriétés du tSampleRow

Famille du composant	Processing	
Fonction	Le composant **tSampleRow** filtre les lignes en fonction de leur position.	
Objectif	**Le tSampleRow** permet de sélectionner des lignes en fonction d'une liste de lignes simples et/ou de groupes de lignes.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Range	Renseignez la position ou l'intervalle choisi, en utilisant la syntaxe appropriée, pour récupérer une liste de lignes simples et/ou de groupes de lignes.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant traite des flux de données, il nécessite donc des composants d'entrée et sortie.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Filtrer des lignes et des groupes de lignes

Ce scénario Java décrit un Job à trois composants. On utilise un composant **tRowGenerator** pour créer des entrées au hasard qui sont directement envoyées au **tSampleRow**, où elles seront filtrées en fonction de leur position. Dans ce scénario, on suppose que le flux d'entrée contient les noms des vendeurs associés au nombre de produits qu'ils ont chacun vendus ainsi que le nombre d'années passées dans l'entreprise. Le résultat de l'opération de filtrage apparaît dans la console **Run**.

- A partir de la **Palette**, cliquez déposez les composants suivants dans l'éditeur graphique : le **tRowGenerator**, le **tSampleRow** et le **tLogRow**.
- Connectez ces trois composants à l'aide de liens de type **Row** > **Main**.

- Dans l'éditeur graphique, sélectionnez le **tRowGenerator**.
- Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tRowGenerator**.
- Dans la vue **Basic settings**, définissez l'option **Schema** en mode **Built-In** puis cliquez sur le bouton **[...]** à côté du champ **Edit Schema** pour définir les données que vous voulez utiliser en entrée. Dans ce scénario, le schéma est composé de cinq colonnes.

- Dans l'onglet **Basic settings**, cliquez sur **RowGenerator Editor** pour définir les données à générer.
- Dans l'éditeur **RowGenerator Editor**, précisez le nombre de lignes à générer dans le champ **Number of Rows for RowGenerator** puis cliquez sur **OK** pour fermer l'éditeur.

- Dans l'éditeur graphique, sélectionnez le **tSampleRow**.
- Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tSampleRow**.

- Dans l'onglet **Basic settings**, définissez l'option **Schema** en mode **Built-In** puis cliquez sur **Sync columns** pour récupérer le schéma à partir du composant **tRowGenerator**.
- Dans la boîte de texte **Range**, définissez le filtre de sélection des lignes en utilisant la syntaxe appropriée comme indiqué plus haut. Dans ce scénario, on veut sélectionner la première et la cinquième ligne, ainsi que le groupe de lignes entre 9 et 12.
- Dans l'espace graphique, sélectionnez le **tLogRow** puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**). Pour plus d'informations sur les propriétés du **tLogRow**, consultez *tLogRow*.
- Enregistrez votre Job puis appuyez sur **F6** pour l'exécuter.

```
Starting job sample_Row at 11:16 20/08/2008.
.--+-------------------+----------+---------+-----------.
|                     tLogRow_1                          |
|=-+-------------------+----------+---------+----------=|
|ID|Years_in_Enterprise|First_Name|Last_Name|Sale_Values|
|=-+-------------------+----------+---------+----------=|
|1 |1                  |Martin    |Taft     |16         |
|5 |1                  |Franklin  |Adams    |45         |
|9 |1                  |William   |Madison  |15         |
|10|1                  |George    |Kennedy  |47         |
|11|1                  |George    |Van Buren|16         |
|12|1                  |Theodore  |Grant    |68         |
'--+-------------------+----------+---------+-----------'

Job sample_Row ended at 11:16 20/08/2008. [exit code=0]
```

Les résultats filtrés apparaissent dans la console de log. Ils affichent la première et la cinquième ligne, ainsi que les lignes entre 9 et 12.

tSortRow

Propriétés du tSortRow

Famille de composant	Processing	
Fonction	Ce composant trie les données d'entrée basées sur une ou plusieurs colonnes, selon un type de tri et un ordre.	
Objectif	Il permet d'établir des métriques et des tables de classification.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Criteria	Cliquez sur **[+]** pour ajouter autant de lignes que nécessaire pour mettre en place le tri. La première colonne définie dans le schéma est sélectionnée par défaut.
		Schema column : Sélectionnez la colonne de votre schéma sur laquelle vous souhaitez baser votre tri. Notez que l'ordre est important car il détermine la priorité de tri.
		Sort type : Numérique ou Alphabétique.
		Order : Ordre ascendant ou descendant.
Advanced settings	*Sort on disk* Indisponible pour la version Map/Reduce du composant.	Personnalisez la mémoire utilisée de manière temporaire pour stocker les données de sortie. **Temp data directory path** : Définissez le chemin d'accès au dossier dans lequel stocker les fichiers temporaires. **Create temp data directory if not exists** : Cochez cette case afin de créer le dossier s'il n'existe pas. **Buffer size of external sort** : Entrez la taille de la mémoire physique à allouer au processus de tri.

	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n'est pas disponible dans la version Map/Reduce de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant manipule le flux de données et par conséquent requiert un composant en entrée et un composant en sortie. Il s'agit donc d'un composant intermédiaire.	
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Pour un scénario présentant un Job Map/Reduce utilisant ce composant, consultez *Scénario 2 : Dédoublonner des entrées en utilisant des composants Map/Reduce*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	

Scénario : Trier des données

Le scénario suivant décrit un Job constitué de trois composants. Un composant **tRowGenerator** est utilisé pour créer des entrées de façon aléatoire. Ces entrées seront ensuite envoyées au composant **tSortRow** afin d'être triées selon une valeur définie. Dans ce scénario, le flux d'entrée contient des noms de vendeurs ainsi que leur volume de vente respectif et leur nombre d'années d'ancienneté dans l'entreprise. Le résultat de l'opération de tri est affiché dans la console **Run**.

- Cliquez et déposez les trois composants requis pour ce scénario : **tRowGenerator**, **tSortRow** et **tLogRow**.
- Connectez-les à l'aide de liens de type **Row Main**.
- Dans le **RowGenerator editor**, définissez les valeurs qui seront créées de manière aléatoire et qui seront ensuite triées par le **tSortRow**. Pour plus d'informations concernant l'utilisation du **tRowGenerator**, consultez *Propriétés du tRowGenerator*.

- Dans ce scénario, chaque vendeur est classé en fonction de la valeur de ses ventes (*Sales*) et de son ancienneté dans l'entreprise (*YearsInComp*).

- Double-cliquez sur **tSortRow** pour afficher l'onglet **Basic settings**. Définissez la priorité de tri sur la valeur des ventes et, en second critère, sur l'ancienneté.

- Utilisez le bouton **[+]** pour ajouter le nombre de lignes de critères requis. Paramétrez le type de tri, dans cet exemple, les deux critères sont de type numérique. Enfin, étant donné que la sortie est une classification, définissez l'ordre de tri comme descendant.

- Dans l'onglet **Advanced settings**, cochez l'option **Sort on disk** pour modifier les paramètres de la mémoire temporaire. Dans le champ **Temp data directory path**, renseignez le chemin d'accès au dossier dans lequel vous voulez stocker les données temporaires. Dans le champ **Buffer size of external sort**, définissez la taille maximale de la mémoire tampon à allouer au traitement des données.

La valeur par défaut de la mémoire tampon est de 1000000 mais plus vous traitez un nombre important de lignes et/ou colonnes, plus cette valeur devra être élevée pour éviter l'interruption automatique du Job signifiée par le message d'erreur "out of memory".

- Assurez-vous que ce flux est connecté au composant de sortie **tLogRow**, afin d'afficher le résultat dans la console **Run**.

- Appuyez sur **F6** pour exécuter le Job ou passez à la vue **Run** et cliquez sur le bouton **Run**. Le classement est d'abord basé sur la valeur des ventes puis sur le nombre d'années d'ancienneté.

Execution
Run
Kill
Clear
Starting job feature_17d10 at 15:2d
21/12/2010.
[statistics] connecting to socket on port
3396
[statistics] connected
10|4|McKinley|94
3|4|Coolidge|86
2|2|Madison|82
4|4|Harding|52
9|2|Kennedy|42
6|4|Cleveland|30
7|2|Harrison|28
1|4|Tyler|19
8|4|Nixon|17
5|4|Ford|17
[statistics] disconnected
Job feature_17d10 ended at 15:2d
21/12/2010. [exit code=0]
Line limit
100

tSplitRow

Propriétés du tSplitRow

Famille de composant	Processing/Fields	
Fonction	Le composant **tSplitRow** découpe une ligne en plusieurs lignes.	
Objectif	Ce composant vous permet de séparer une ligne d'entrée en plusieurs lignes de sortie.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**) , soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Columns mapping	Cliquez sur le bouton **[+]** pour ajouter autant de lignes que nécessaire pour effectuer le mapping des colonnes d'entrée vers les colonnes de sortie.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	

Utilisation	Ce composant découpe une ligne d'entrée en de multiples lignes de sortie, en mappant les colonnes d'entrée vers les colonnes de sortie.

Scénario 1 : Séparer une ligne en plusieurs lignes

Ce scénario décrit un Job a trois composants. Une ligne de données contenant des informations concernant deux entreprises sera séparée en deux lignes.

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tFixedFlowInput**, **tSplitRow** et **tLogRow**.

2. Reliez-les à l'aide de liens **Row Main**.

3. Double-cliquez sur le **tFixedFlowInput** afin d'ouvrir sa vue **Basic settings**.

4. Sélectionnez **Use Inline Content(delimited file)** dans la zone **Mode**.

5. Dans la zone **Content**, saisissez les scripts suivants :

   Talend;LA;California;537;5thAvenue;IT;Lionbridge;Memphis;Tennessee;537;Lincoln Road;IT Service;

6. Cliquez sur **Edit schema** pour ouvrir une boîte de dialogue afin d'éditer le schéma des données d'entrée.

7. Cliquez sur le bouton **[+]** pour ajouter douze colonnes d'entrée : *Company*, *City*, *State*, *CountryCode*, *Street*, *Industry*, *Company2*, *City2*, *State2*, *CountryCode2*, *Street2* et *Industry2*.

8. Cliquez sur **OK** pour fermer la boîte de dialogue.

9. Double-cliquez sur le composant **tSplitRow** afin d'ouvrir sa vue **Basic settings**.

10. Cliquez sur **Edit schema** pour configurer le schéma des données de sortie.

11.Cliquez sur le bouton **[+]** sous la table *tSplitRow_1(Output)* affin d'ajouter quatre colonnes de sortie : *Company*, *CountryCode*, *Address* et *Industry*.

12.Cliquez sur **OK** pour fermer la boîte de dialogue. Une table vide contenant le nom des colonnes définies dans l'étape précédente apparaît dans la zone **Columns mapping** :

13.Cliquez sur le bouton **[+]** sous la table **Columns mapping** afin d'ajouter deux lignes de sortie.

14.Renseignez la table **Columns mapping** en saisissant les valeurs suivantes dans les colonnes :

Company: *row1.Company*, *row1.Company2*;

Country: *row1.CountryCode*, *row1.CountryCode2*;

Address: *row1.Street+","+row1.City+","+row1.State*, *row1.Street2+","+row1.City2+","+row1.State2*;

Industry: *row1.Industry*, *row1.Industry2*;

La valeur dans la colonne *Address*, par exemple, *row1.Street+","+row1.City+","+row1.State*, affiche une adresse absolue en combinant les valeurs des colonnes *Street*, *City* et *State*. "*row1*" utilisé dans les valeurs de chaque colonne fait référence à la ligne d'entrée du **tFixedFlowInput**.

15.Double-cliquez sur le **tLogRow** afin d'ouvrir sa vue **Basic settings**.

16.Cliquez sur le bouton **Sync columns** pour récupérer le schéma défini dans le composant précédent.

17.Sélectionnez **Table (print values in cells of a table)** dans la zone **Mode**.

18.Sauvegardez le Job et appuyez sur **F6** pour l'exécuter.

```
[statistics] connecting to socket on port 3706
[statistics] connected
.----------+-----------+-----------------------------+----------.
|                          tLogRow_1                              |
|=---------+-----------+-----------------------------+---------=|
|Company   |CountryCode|Address                      |Industry  |
|=---------+-----------+-----------------------------+---------=|
|Talend    |537        |5thAvenue,LA,California      |IT        |
|Lionbridge|537        |Lincoln Road,Memphis,Tennessee|IT Service|
'----------+-----------+-----------------------------+----------'
[statistics] disconnected
Job Split ended at 16:21 27/10/2011. [exit code=0]
```

Les données d'entrée en une ligne sont séparées en deux lignes différentes, chaque ligne contenant es informations d'une des deux entreprises.

tWriteJSONField

Propriétés du tWriteJSONField

Famille de composant	Processing/Fields	
Fonction	Le composant **tWriteJSONField** transforme les données entrantes en champs JSON. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job Map/Reduce **Talend** afin de générer du code Map/Reduce. Pour plus d'informations, consultez, *Le tFileOutputJSON dans des Jobs Map/Reduce Talend*. Dans ce cas, le **tWriteJSONField** appartient à la famille de composant MapReduce.	
Objectif	Le **tWriteJSONField** transforme les données entrantes en champs JSON et les transfère dans un fichier, une table de base de données.	
Basic settings	*Output Column*	Liste des colonnes définies dans le schéma de sortie contenant les champs JSON générés.
	Configure JSON Tree	Cliquez pour ouvrir l'interface de création de la structure de données JSON. Pour plus d'informations, consultez *Configurer une arborescence JSON*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Cliquez sur ce bouton pour synchroniser le schéma du fichier de sortie et le schéma du fichier d'entrée. La fonction **Sync** s'affiche uniquement lorsque le lien **Row** est connecté au composant de sortie.
	Group by	Définissez l'ensemble d'agrégation, les colonnes que vous souhaitez utiliser pour regrouper les données. *Assurez-vous que les données à grouper sont un ordre séquentiel.*
	Remove root node	Cochez cette case pour supprimer le nœud racine du champ JSON généré.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n'est pas disponible dans la version Map/Reduce de ce composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.	

	NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Précédé d'un composant d'entrée, ce composant met les données d'entrée dans un champ JSON.
Utilisation dans les Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.

Configurer une arborescence JSON

Lorsque vous configurez une arborescence JSON, le type d'élément par défaut est *string*. Si un élément n'est pas de type *string*, vous devez ajouter un attribut à l'élément afin de définir son type.

- Pour un élément de type *integer*, *double*, *float* ou *boolean*, vous devez ajouter un attribut nommé **type** et définir sa valeur, dans la colonne **Static Value**, en **integer**, **number**, **float**, ou **boolean** respectivement.
- Pour un élément de type *array*, vous devez ajouter un attribut nommé **class** et définir sa valeur, dans la colonne **Static Value**, en **array** puis ajouter un sous élément nommé **element** défini en élément de boucle (**loop element**).
- Pour un élément de type *object*, vous devez ajouter un attribut nommé **class** et définir sa valeur, dans la colonne **Static Value**, en **object**.

L'image suivante montre un exemple de configuration d'arborescence JSON :

Scénario : Ecrire des données plates dans des champs JSON

Dans ce scénario, les données plates sont mises dans des champs JSON via le composant **tWriteJSONField**.

Relier les composants

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tFixedFlowInput**, un **tWriteJSONField** et un **tLogRow**.
2. Reliez le **tFixedFlowInput** au **tWriteJSONField** à l'aide d'un lien **Row** > **Main**.
3. Reliez le **tWriteJSONField** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

Configurer les composants

1. Double-cliquez sur le **tFixedFlowInput** pour afficher sa vue **Basic settings**.

2. Cliquez sur **Edit schema** pour ouvrir l'éditeur du schéma.

Cliquez sur le bouton **[+]** pour ajouter trois colonnes, respectivement nommée *firstname*, *lastname* et *dept*, de type **String**.

Cliquez sur **OK** pour fermer l'éditeur.

3. Sélectionnez l'option **Use Inline Content** et saisissez les données ci-dessous dans le champ **Content** :

```
Andrew;Wallace;Doc
John;Smith;R&D
Christian;Dior;Sales
```

4. Cliquez sur le **tWriteJSONField** pour afficher sa vue **Basic settings**.

Cochez la case **Remove root node** afin de supprimer le nœud racine des champs JSON générés.

5. Cliquez sur le bouton **[...]** à côté du champ **Configure JSON Tree** pour ouvrir l'éditeur de l'arborescence XML.

Le schéma du **tFixedFlowInput** apparaît dans le panneau **Linker source**.

6. Dans le panneau **Linker target**, cliquez sur le nœud racine par défaut (**rootTag**) et saisissez *staff*, le nom du nœud racine du champ JSON à générer.

7. Cliquez-droit sur *staff* et sélectionnez **Add Sub-element** dans le menu contextuel.

8. Dans la boîte de dialogue, saisissez le nom du sous-nœud, *firstname*.

Répétez les étapes précédentes afin d'ajouter deux autres sous-nœuds, respectivement nommés *lastname* et *dept*.

9. Cliquez-droit sur **firstname** et sélectionnez **Set As Loop Element** dans le menu contextuel.

10. Déposez la ligne *firstname* du panneau **Linker source** dans la ligne du même nom, dans le panneau **Linker target**.

    Dans la boîte de dialogue, sélectionnez **Add linker to target node**.

    Cliquez sur **OK** pour fermer la boîte de dialogue.

    Répétez ces étapes afin de lier les deux autres éléments.

    Cliquez sur **OK** pour fermer l'éditeur d'arborescence XML.

11. Cliquez sur **Edit schema** pour ouvrir l'éditeur du schéma.

12. Cliquez sur le bouton **[+]** du panneau de droite pour ajouter une colonne, nommée *staff*, qui contiendra les données JSON générées.

    Cliquez sur **OK** pour fermer l'éditeur.

13. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.

    Sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

Exécuter le Job

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder votre Job.

2. Cliquez sur **F6** pour l'exécuter.

```
.-------------------------------------------------------------
|                          tLogRow_1
|=------------------------------------------------------------=
|staff
|=------------------------------------------------------------=
|{"first_name":"Andrew","last_name":"Wallace","dept":"Doc"}
|{"first_name":"John","last_name":"Smith","dept":"R&D"}
|{"first_name":"Christian","last_name":"Dior","dept":"Sales"}
'-------------------------------------------------------------
```

    Comme affiché ci-dessus, les champs JSON ont été correctement générés et les paramètres du nœud racine ont bien été supprimés.

Scénarios associés

Pour des scénarios associés, consultez

- *Scénario : Récupérer les messages d'erreur lors de l'extraction de données de champs JSON*.
- *Scénario : Extraire la structure d'un fichier XML et l'insérer dans les champs d'une base de données*.
- *Mapping de données XML*.

tXMLMap

Propriétés du tXMLMap

Famille de composant	Processing/XML	
Fonction	Le composant **tXMLMap** est un composant avancé personnalisable permettant de transformer et de router des flux de données XML (données de type **document**), particulièrement pour traiter de nombreuses sources de données XML, avec ou sans jointure sur les données plates.	
Objectif	Le **tXMLMap** transforme et route des données à partir d'une ou plusieurs source(s) de données vers une ou plusieurs destination(s).	
Basic settings	*Map editor*	Le Map Editor vous permet de définir les propriétés du routage et des transformations du **tXMLMap**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Plusieurs utilisations sont possibles, de la simple réorganisation des champs de données aux transformations les plus complexes, telles que le multiplexage et le démultiplexage de données, la concaténation, l'inversion, le filtrage, etc. Lorsque c'est nécessaire, vous pouvez définir vos sorties pour les flux XML, à l'aide de l'élément "**group**", de l'élément "**aggregate**", d'éléments vides et de toute autre fonctionnalité, comme **All in one**. Pour plus d'informations concernant ces fonctionnalités, consultez le *Guide utilisateur du Studio Talend*. Ce composant est utilisé comme composant intermédiaire et correspond parfaitement au processus nécessitant de nombreuses sources de données XML, comme les processus ESB de requête/réponse.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	
Limitation	Les limites sont les suivantes : - L'utilisation du composant **tXMLMap** requiert des connaissances minimales en Java et XML afin d'exploiter au mieux ses fonctionnalités. - Ce composant est une étape de jonction. Il ne peut donc être ni un composant d'entrée, ni un composant de sortie. - Au moins un élément de boucle est requis pour chaque flux de données XML.	

Les sections suivantes présentent plusieurs scénarios génériques concernant l'utilisation du composant **tXMLMap**. Si vous souhaitez consulter des exemples spécifiques d'utilisation de ce composant avec les composants ESB afin de construire des services de données, consultez les scénarios des composants suivants :

- *Scénario 2 : Utiliser le tEBSConsumer avec des en-têtes SOAP personnalisés*.
- *Scénario : Effectuer une requête sur les nom d'aéroports à partir des codes pays*.
- *Scénario : Envoi d'un message via un service sans attente de réponse*.
- *Scénario : Retourner une réponse "Hello world"*.

Si vous souhaitez plus d'informations concernant les principes du mapping de multiples flux d'entrée et de sortie, consultez le *Guide utilisateur du Studio Talend*.

Scénario 1 : Mapper et transformer des données de source XML

Dans ce scénario, un Job à trois composants est exécuté pour mapper et transformer des données provenant d'une source XML, *customer.xml*, et générer un flux de sortie XML pouvant être réutilisé dans différents buts, comme, par exemple, pour une requête ESB.

Ces trois composants sont :

- **tFileInputXML** : ce composant est utilisé pour fournir des données d'entrée au **tXMLMap**.
- **tXMLMap** : ce composant mappe et transforme les flux de données XML en un seul flux de données XML.
- **tLogRow** : ce composant est utilisé pour afficher les données de sortie.

Afin de reproduire ce scénario, procédez comme dans les sections suivantes :

Déposer et relier les composants

Pour ce faire, procédez comme suit :

1. De la **Palette**, déposez un **tFileInputXML**, un **tXMLMap** et un **tLogRow** dans l'espace de modélisation graphique.

   Un composant utilisé dans l'espace de modélisation peut être renommé comme vous le souhaitez. Dans ce scénario, le composant d'entrée est nommé *Customers*. Pour plus d'informations concernant le renommage d'un composant, consultez le *Guide utilisateur du Studio Talend*.

2. Cliquez-droit sur le composant **tFileInputXML** nommé *Customers* pour ouvrir son menu contextuel.
3. Dans ce menu, sélectionnez **Row** > **Main** afin de relier ce composant au **tXMLMap**.
4. Répétez l'opération afin de connectez ensuite le **tXMLMap** au **tLogRow** à l'aide d'un lien **Row** > ***New Output* (Main)**. Une boîte de dialogue s'ouvre et vous demande de nommer ce lien de sortie. Dans ce scénario, nommez-le *Customer*.

Vous pouvez ensuite continuer à configurer vos composants.

Configurer le flux d'entrée

Pour ce faire, procédez comme suit :

1. Double-cliquez sur le composant **tFileInputXML** nommé *Customers* afin d'afficher sa vue **Basic settings**.

2. A côté du champ **Edit schema**, cliquez sur le bouton **[...]** afin d'ouvrir l'éditeur du schéma.

3. Dans l'éditeur du schéma, cliquez sur le bouton **[+]** pour ajouter une ligne.

4. Dans la colonne **Column**, saisissez un nouveau nom pour cette ligne. Dans ce scénario, le nom est *Customer*.

5. Dans la colonne **Type**, sélectionnez le type de données de cette ligne. Dans ce scénario, le type est *Document*. Le type de données "document" est essentiel pour utiliser au maximum les fonctionnalités du **tXMLMap**. Pour plus d'informations concernant ce type de données, consultez le *Guide utilisateur du Studio Talend*.

6. Cliquez sur **OK** pour valider ce changement. Une ligne est ajoutée automatiquement à la table **Mapping**.

7. Dans le champ **File name/Stream**, parcourez votre système jusqu'au fichier XML source fournissant les données client, ou bien saisissez son chemin d'accès. Dans ce scénario, saisissez *"E:/Studio/Customer.xml"*.

8. Dans le champ **Loop XPath query**, saisissez `"/"` pour remplacer le paramètre par défaut. Cela signifie que les données source sont interrogées depuis la racine.

9. Dans la colonne **XPath query** de la table **Mapping**, saisissez le XPath. Dans ce scénario, saisissez `".."`, ce qui signifie que toutes les données de la source seront interrogées.

10. Cochez la case **Get Nodes** dans la colonne de la table **Mapping**.

Afin de construire le flux de données de type **Document**, il est nécessaire d'obtenir les nœuds de ce composant.

Configurer le tXMLMap pour effectuer des transformations

Pour ce faire, procédez comme suit :

1. Double-cliquez sur le composant **tXMLMap** pour ouvrir le **Map Editor**.

   Notez que la zone d'entrée est déjà remplie par les tables d'entrée définies et que la table supérieure est la table principale d'entrée (flux **Main**).

2. Dans la table gauche, cliquez-droit sur **Customer** pour ouvrir le menu contextuel.

3. Dans ce menu contextuel, sélectionnez **Import From File** et dans la fenêtre pop-up, parcourez votre système jusqu'au fichier source correspondant, afin d'importer la structure XML utilisée par les données reçues par le **tXMLMap**. Dans ce scénario, le fichier source est *Customer.xml*, qui est également connectés au **tFileInputXML** (*Customers*).

   Vous pouvez également importer une arborescence XML à partir d'un fichier XSD. Lors de l'import d'une structure XML d'entrée ou de sortie depuis un fichier XSD, vous pouvez choisir un élément pour en faire la racine de votre structure XML. Pour plus d'informations concernant l'import d'une arborescence XML à partir d'un fichier XSD, consultez le *Guide utilisateur du Studio Talend*.

4. Dans la table du côté gauche, cliquez-droit sur le nœud *Customer* et sélectionnez **As loop element** pour le définir comme élément de boucle.

5. Dans la partie inférieure du **Map Editor**, cliquez sur l'onglet **Schema editor** afin d'afficher la vue correspondante.

6. Du côté droit de cette vue, cliquez sur le bouton **[+]** pour ajouter une ligne à la table *Customer* et renommez cette ligne *Customer_States*.

7. Dans la colonne **Type** de la ligne *Customer_States*, sélectionnez **Document** comme type de données. La racine XML correspondante est ajoutée automatiquement à la table supérieure, du côté droit, qui représente le flux de sortie.

8. Du côté droit de la table supérieure nommée *Customer*, importez la structure XML des données que vous devez utiliser du fichier source XML correspondant. Dans ce scénario, le fichier est *Customer_State.xml*.

9. Cliquez-droit sur le nœud *customer* et sélectionnez **As loop element** dans le menu contextuel.

   Vous pouvez ensuite commencer à mapper le flux d'entrée vers le flux de sortie.

10. Dans la table du côté de l'entrée (gauche) du **Map Editor**, cliquez sur le nœud *id* et glissez-le dans la colonne **Expression** dans la ligne correspondant à la ligne de sortie que vous devez mapper. Dans ce scénario, le nœud est *@id*.

11. Répétez l'opération pour mapper *CustomerName* à *CustomerName*, *CustomerAddress* à *CustomerAddress* et *idState* à *idState* du côté gauche au côté droit.

    Dans un projet réel, vous pouvez avoir besoin de laisser des éléments vides dans votre arborescence XML de sortie. Dans ce cas, vous pouvez utiliser le **tXMLMap** pour les gérer. Pour plus d'informations concernant la gestion des éléments vides via le **tXMLMap**, consultez le *Guide utilisateur du Studio Talend*.

12. Si nécessaire pour générer plusieurs flux XML, cliquez sur la clé anglaise en haut à droite afin d'ouvrir le panneau de paramétrage du mapping et configurer la fonctionnalité **All in one** comme **true**. Dans cet exemple, cette option est sur **true**. Pour plus d'informations concernant la fonctionnalité **All in one**, consultez le *Guide utilisateur du Studio Talend*.

13. Cliquez sur **OK** pour valider le mapping et fermer le **Map Editor**.

    Si vous fermez le **Map Editor** sans avoir défini les éléments de boucle nécessaires, comme expliqué plus tôt dans ce scénario, l'élément racine est automatiquement défini comme élément de boucle.

    Vous pouvez exécuter le Job.

Configurer le tLogRow pour afficher les informations client

1. Double-cliquez sur le composant **tLogRow**pour ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.

```
[statistics] connecting to socket on port 4079
[statistics] connected
<?xml version="1.0" encoding="UTF-8"?>
<customers><customer id="1"><CustomerName>Griffith Paving and
Sealcoatin</CustomerName><CustomerAddress>talend@apres91</CustomerAddress><id
State>7</idState></customer><customer id="56"><CustomerName>Glenn Oaks Office
Supplies</CustomerName><CustomerAddress>1859 Green Bay
Rd.</CustomerAddress><idState>7</idState></customer><customer
id="2"><CustomerName>Bill's Dive Shop</CustomerName><CustomerAddress>511
Maple Ave. Apt. 1B</CustomerAddress><idState>35</idState></customer><customer
id="61"><CustomerName>DBN Bank</CustomerName><CustomerAddress>456 Grossman
Ln.</CustomerAddress><idState>35</idState></customer><customer
id="62"><CustomerName>BBQ Smith's Tex Mex</CustomerName><CustomerAddress>1101
Deerfield Place</CustomerAddress><idState>60</idState></customer><customer
id="63"><CustomerName>Pivot Point College</CustomerName><CustomerAddress>1547
Knolwood Rd.</CustomerAddress><idState>9</idState></customer></customers>
[statistics] disconnected
```

Les informations clients transformées sont affichées dans la console.

Scénario 2 : Créer un flux Lookup pour effectuer une jointure sur des données complémentaires XML

Basé sur le scénario précédent, ce scénario explique comment utiliser un flux Lookup (de référence) dans un flux XML afin d'effectuer une jointure sur les données souhaitées dans le flux XML donné. Les données XML de référence sont contenues dans le fichier *USStates.xml*.

Pour ce faire, un nouveau composant **tFileInputXML** est ajouté au Job précédent afin de charger et d'envoyer des données complémentaires au **tXMLMap**. Le Job doit ressembler à ceci :

Pour reproduire ce scénario, procédez comme dans les sections suivantes.

Configurer le flux de données pour le flux de référence

Pour ce faire, procédez comme suit :

1. De la **Palette**, déposez un composant **tFileInputXML** dans l'espace de modélisation graphique.

   Un composant utilisé dans l'espace de modélisation peut être renommé comme vous le souhaitez. Dans ce scénario, le composant ajouté est nommé *USStates*. Pour plus d'informations concernant le renommage d'un composant, consultez le *Guide utilisateur du Studio Talend*.

2. Double-cliquez sur le composant **tFileInputXML** nommé *USStates* pour ouvrir son menu contextuel et sélectionnez **Row** > **Main** afin de relier ce composant au **tXMLMap**. Puisque cette connexion **Row** est la seconde que vous créez, elle est de type **Lookup**.

3. Double-cliquez sur le composant **tFileInputXML** nommé *USStates* pour ouvrir sa vue **Basic settings**.

4. A côté du champ **Edit schema**, cliquez sur le bouton **[...]** pour ouvrir l'éditeur du schéma.

5. Dans l'éditeur, cliquez sur le bouton **[+]** pour ajouter une ligne et renommez-la *USState*.

6. Dans la colonne **Type**, sélectionnez **Document** dans la liste déroulante.

7. Cliquez sur **OK** pour valider les modifications.

8. Dans le champ **File name/Stream**, parcourez votre système ou saisissez le chemin d'accès au fichier *USStates.xml*.

   Les schémas d'entrée peuvent être stockés dans le nœud **Metadata** de la vue **Repository** pour que vous puissiez les retrouver facilement. Pour plus d'informations concernant la création de métadonnées dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

9. Dans le champ **Loop XPath query**, saisissez `"/"` afin de remplacer la valeur par défaut. Cela signifie que la boucle est basée sur la racine.

10. Dans la table **Mapping**, où une ligne est déjà ajoutée automatiquement, saisissez `" . "` dans la colonne **XPath query**, afin de récupérer les Etats des Etats-Unis depuis le fichier source.

11. Dans la colonne **Get Nodes**, cochez la case. Cela va récupérer la structure XML des données de type **Document**.

Configurer la transformation

Pour ce faire, procédez comme suit :

1. Double-cliquez sur le composant **tXMLMap** pour ouvrir le **Map Editor**.

   Notez que la zone d'entrée est déjà remplie par les tables d'entrée définies et que la table supérieure est la table principale d'entrée (flux **Main**).

2. Cliquez-droit sur **USState** dans la table **lookup** pour ouvrir son menu contextuel.

3. Dans le menu contextuel, sélectionnez **Import From File** et, dans la boîte de dialogue qui s'ouvre, parcourez votre système jusqu'au fichier source correspondant afin d'importer la structure XML utilisée par les données que doit recevoir le **tXMLMap**. Dans ce scénario, le fichier source est *USStates.xml*, c'est-à-dire la source des données du composant **tFileInputXML** nommé *USStates*.

4. Cliquez-droit sur le nœud **States** et sélectionnez **As loop element** dans le menu contextuel.

5. Dans la table principale d'entrée, cliquez-droit sur le nœud *idState* et déposez-le, dans la table **lookup**, dans la colonne **Exp.key** correspondant à la ligne *idState*. Cela crée une jointure entre les deux tables sur les données *idState*, parmi lesquelles le nœud *idState* du flux principal fournit la clé de référence (**lookup key**).

Vous pouvez commencer à modifier le mapping du scénario précédent afin d'effectuer une jointure sur les données plates du flux d'entrée. Le mapping doit ressembler à ceci :

6. Dans la table **lookup** de gauche dans l'éditeur, cliquez sur la ligne *LabelState* et déposez-la sur le nœud *customer* de la sortie. Une boîte de dialogue s'ouvre.

7. Dans cette boîte de dialogue, sélectionnez **Create as sub-element of target node** puis cliquez sur **OK**. Cette opération ajoute un sous-élément dans l'arborescence XML de sortie et le mappe également avec *LabelState* dans l'entrée.

8. Si nécessaire pour générer plusieurs flux XML, cliquez sur la clé anglaise en haut à droite afin d'ouvrir le panneau de paramétrage du mapping et configurer la fonctionnalité **All in one** comme **true**. Dans cet exemple, cette option est sur **true**. Pour plus d'informations concernant la fonctionnalité **All in one**, consultez le *Guide utilisateur du Studio Talend*.

9. Cliquez sur **OK** pour valider le mapping et fermer le **Map Editor**.

Exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** pour exécuter le Job.

   La console de la vue **Run** affiche les résultats d'exécution comme suit :

```
[statistics] connecting to socket on port 3854
[statistics] connected
<?xml version="1.0" encoding="UTF-8"?>
<customers><customer id="1"><CustomerName>Griffith Paving and
Sealcoatin</CustomerName><CustomerAddress>talend@apres91</Custo
merAddress><idState>7</idState><LabelState>Connecticut</LabelSt
ate></customer><customer id="56"><CustomerName>Glenn Oaks
Office Supplies</CustomerName><CustomerAddress>1859 Green Bay
Rd.</CustomerAddress><idState>7</idState><LabelState>Connecticu
t</LabelState></customer><customer id="2"><CustomerName>Bill's
Dive Shop</CustomerName><CustomerAddress>511 Maple Ave. Apt.
1B</CustomerAddress><idState>35</idState><LabelState>Ohio</Labe
lState></customer><customer id="61"><CustomerName>DBN
Bank</CustomerName><CustomerAddress>456 Grossman
Ln.</CustomerAddress><idState>35</idState><LabelState>Ohio</Lab
elState></customer><customer id="62"><CustomerName>BBQ Smith's
Tex Mex</CustomerName><CustomerAddress>1101 Deerfield
Place</CustomerAddress><idState>60</idState><LabelState/></cust
omer><customer id="63"><CustomerName>Pivot Point
College</CustomerName><CustomerAddress>1547 Knolwood
Rd.</CustomerAddress><idState>9</idState><LabelState>Florida</L
abelState></customer></customers>
[statistics] disconnected
```

   Les noms des états des Etats-Unis correspondant à l'ID défini, fourni comme clé de référence par le flux de données principal, sont sélectionnés et affichés en sortie.

Un tutoriel étape par étape relatif aux jointures est disponible sur le site de la communauté technique Talend. Pour plus d'informations, consultez http://www.talendforge.org/tutorials/tutorial.php?language=french&idTuto=101.

Scénario 3 : Mapper des données à l'aide d'un filtre

Ce scénario se base sur *Scénario 2 : Créer un flux Lookup pour effectuer une jointure sur des données complémentaires XML* et explique comment appliquer une (des) condition(s) de filtre pour sélectionner les données qui vous intéressent, à l'aide du **tXMLMap**.

Par exemple, vous souhaitez sélectionner les donnée clients pour lesquelles l'ID de l'état est *9*.

Pour reproduire ce scénario, procédez comme suit :

1. Dans votre Studio, ouvrez le Job utilisé dans le scénario précédent afin de l'afficher dans l'espace de modélisation graphique.

2. Double-cliquez sur le **tXMLMap** pour ouvrir son éditeur. Dans cet éditeur, les flux d'entrée et de sortie sont déjà mappés puisque le Job est répliqué du scénario précédent.

3. Du côté de la sortie (à droite), cliquez sur le bouton pour ouvrir la zone de filtre.

4. Dans cette zone de filtre, déposez le nœud *idState* du flux d'entrée. Le chemin Xpath de l'*idState* est ajouté automatiquement dans cette zone de filtre.

5. Toujours dans cette zone, saisissez votre condition de filtre en Java. Dans ce scénario, la condition est la suivante : `"9".equals([row1.Customer:/Customers/Customer/Address/idState])`

6. Si nécessaire pour générer plusieurs flux XML, cliquez sur la clé anglaise en haut à droite afin d'ouvrir le panneau de paramétrage du mapping et configurer la fonctionnalité **All in one** comme **true**. Dans cet exemple, cette option est sur **true**. Pour plus d'informations concernant la fonctionnalité **All in one**, consultez le *Guide utilisateur du Studio Talend*.

7. Cliquez sur **OK** pour valider les modifications et fermer l'éditeur.

8. Appuyez sur **F6** pour exécuter ce Job.

   Les résultats d'exécution s'affichent dans la vue **Run** :

```
[statistics] connecting to socket on port 3898
[statistics] connected
<?xml version="1.0" encoding="UTF-8"?>
<customers><customer id="63"><CustomerName>Pivot Point
College</CustomerName><CustomerAddress>1547 Knolwood
Rd.</CustomerAddress><idState>9</idState><LabelState>Florida</LabelState>
</customer></customers>
[statistics] disconnected
```

   Les résultats montrent que le client *Pivot Point College* est sélectionné, puisque l'ID de son état est *9*, ce qui représente la Floride dans ce scénario.

Scénario 4 : Capturer les données rejetées par le flux Lookup et le filtre

Les données rejetées par le flux **Lookup** et les conditions de filtre définies dans le **tXMLMap** peuvent être capturées et écrites en sortie par le composant lui-même.

Ce scénario se base sur *Scénario 3 : Mapper des données à l'aide d'un filtre*, et explique comment capturer les données rejetées par le flux Lookup et les filtrer défini dans les sections précédentes.

Dans ce scénario, un autre composant **tLogRow** est ajouté au Job du scénario précédent. Le Job ressemble à ceci :

Pour reproduire ce scénario, procédez comme suit :

1. Dans le Studio, ouvrez le Job utilisé dans le scénario précédent dans l'espace de modélisation graphique.

2. De la **Palette**, déposez le composant **tLogRow** dans l'espace de modélisation graphique.

3. Cliquez sur le **tXMLMap** pour ouvrir son menu contextuel. Dans ce menu, sélectionnez **Row > *New Output* (Main)** pour connecter ce composant au nouveau **tLogRow**. Une boîte de dialogue s'ouvre et vous demande de nommer ce lien de sortie. Dans ce scénario, nommez le lien *Reject*.

4. Cliquez sur **OK** pour valider la création.

5. Double-cliquez sur le composant **tXMLMap** pour ouvrir son éditeur. Une table *Reject* vide a été ajoutée à la sortie pour représenter le flux de sortie contenant les données rejetées. Vous devez compléter cette table afin que l'éditeur ressemble à ceci :

6. Sélectionnez la table vide *Reject*.

7. Dans la partie inférieure de l'éditeur, cliquez sur l'onglet **Schema editor** pour ouvrir la vue correspondante.

8. Dans la partie droite de la vue **Schema editor**, cliquez sur le bouton **[+]** pour ajouter à la table les lignes que vous souhaitez utiliser. Dans ce scénario, cliquez quatre fois pour ajouter quatre lignes à la table *Reject*.

9. Dans la table *Reject* à droite de la vue **Schema editor**, renommez les lignes ajoutées. Renommez-les respectivement : *ID*, *Customer*, *idState*, et *LabelState*.

Dans ce scénario, les données du flux de sortie sont des données plates. Cependant, vous pouvez créer une arborescence XML à l'aide du type de données **document**. Pour plus d'informations concernant l'utilisation du type **document**, consultez *Scénario 1 : Mapper et transformer des données de source XML*.

La table *Reject* est complétée et vous avez défini le schéma du flux de sortie utilisé pour capturer les données rejetées. Vous devez ensuite mapper le flux de données et configurer la (les) condition(s) pour capturer les données rejetées qui vous intéressent.

Configurer le mapping et exécuter le Job

1. Dans la table d'entrée principale, à gauche de l'éditeur, glissez-déposez le nœud *id* dans la colonne **Expression** de la ligne correspondante de la table de sortie *Reject*.
2. Glissez-déposez *CustomerName*, *CustomerAddress* et *idState* de la table d'entrée principale et *LabelState* de la table **lookup** dans les lignes correspondantes de la table de sortie *Reject*.
3. Dans la partie supérieure de la sortie dans le **Map editor**, sélectionnez la table *Reject*.
4. En haut de cette table, cliquez sur le bouton pour ouvrir la zone de configuration.

5. Dans la ligne **Catch Output Reject** de la zone de configuration, sélectionnez **true** dans la liste déroulante. Ainsi, le **tXMLMap** écrit les données rejetées par le filtre défini pour le flux de sortie *Customer* dans le scénario précédent.
6. Répétez l'opération pour sélectionner **true** dans la ligne **Catch Lookup Inner Join Reject**.
7. Cliquez sur **OK** pour valider vos modifications et fermer l'éditeur.
8. Double-cliquez sur le second composant **tLogRow** pour ouvrir sa vue **Basic settings**.
9. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
10. Appuyez sur **F6** pour exécuter le Job.

Les données capturées rejetées par le flux **Lookup** et le filtre s'affichent comme suit dans la vue **Run** :

```
1|Griffith Paving and Sealcoatin|7|Connecticut
56|Glenn Oaks Office Supplies|7|Connecticut
2|Bill's Dive Shop|35|Ohio
61|DBN Bank|35|Ohio
62|BBQ Smith's Tex Mex|60|
```

Les états de ces clients n'ont pas l'ID *9*. L'ID de l'état du client *BBQ Smith's Tex Mex* est *60*. Ce nombre n'existe pas dans la colonne *idState* du fichier *USState.txt* pour laquelle le Lookup défini a été effectué. Ainsi, les données de ce client sont rejetées par le flux Lookup et les autres données sont rejetées par le filtre.

Les données sélectionnées par le filtre défini dans le scénario précédent s'affiche comme suit au format XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<customers><customer id="63"><CustomerName>Pivot Point
College</CustomerName><CustomerAddress>1547 Knolwood
Rd.</CustomerAddress><idState>9</idState><LabelState>Florida</LabelState>
</customer></customers>
```

Scénario 5 : Mapper des données à l'aide d'un élément "group"

Ce scénario se base sur le Job utilisé dans *Scénario 2 : Créer un flux Lookup pour effectuer une jointure sur des données complémentaires XML* et explique comment définir un élément en tant que **group element** dans le **Map editor** du **tXMLMap** afin de regrouper les données de sortie.

Pour reproduire ce scénario, vous pouvez réutiliser le Job dans *Scénario 2 : Créer un flux Lookup pour effectuer une jointure sur des données complémentaires XML*

Dans ce Job, double-cliquez sur le **tXMLMap** afin d'ouvrir le **Map Editor**.

L'objectif de ce scénario est de regrouper les IDs des clients et les informations relatives aux clients, selon l'Etat dans lequel ils résident. Pour ce faire, vous devez ajuster la structure XML, en prenant en compte les facteurs suivants :

- Les éléments relatifs à l'ID et au nom des clients doivent être situés sous l'élément de boucle. Ils sont donc des sous-éléments de la boucle.
- L'élément de boucle et ses sous-éléments doivent dépendre directement de l'élément de groupe.
- L'élément relatif à l'Etat utilisé comme condition de groupe doit dépendre directement de l'élément de groupe.
- L'élément de groupe ne peut pas être l'élément racine.

Pour prendre en compte un **group element**, les données XML à traiter doivent avoir été triées, par exemple via vos outils XML, à l'aide de l'élément que vous souhaitez utiliser comme condition de regroupement. La capture d'écran ci-dessous présente une partie de la source des données triées. Les clients ayant le même *idState* sont déjà regroupés.

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<Customers>
    <Customer RegisterTime="2001-01-17 06:26:40.000">
        <Name>
            <id>1</id>
            <CustomerName>Griffith Paving and Sealcoatin</CustomerName>
        </Name>
        <Address>
            <CustomerAddress>talend@apres91</CustomerAddress>
            <idState>7</idState>
        </Address>
        <Revenue>
            <Suml>67852</Suml>
        </Revenue>
    </Customer>
    <Customer RegisterTime="1987-02-23 17:33:20.000">
        <Name>
            <id>56</id>
            <CustomerName>Glenn Oaks Office Supplies</CustomerName>
        </Name>
        <Address>
            <CustomerAddress>1859 Green Bay Rd.</CustomerAddress>
            <idState>7</idState>
        </Address>
        <Revenue>
            <Suml>1225.</Suml>
        </Revenue>
    </Customer>
    <Customer RegisterTime="2002-06-07 09:40:00.000">
        <Name>
            <id>2</id>
            <CustomerName>Bill's Dive Shop</CustomerName>
```

A partir de cette analyse, la structure des données de sortie doit se présenter comme suit :

Dans la capture d'écran, le nœud *customers* est la racine, l'élément *Customer* est le *group element* et les données de sortie sont regroupées selon l'élément *LabelState*.

Pour définir un élément de groupe, deux restrictions doivent être respectées :

- Le nœud ne peut être défini comme **group element** ;
- la définition du **group element** doit se faire sur le nœud père de l'élément de boucle.

Une fois l'élément de groupe défini, le premier élément (en-dehors de celui de boucle) est utilisé comme condition de groupe des données de sortie.

Pour effectuer l'ajustement selon cette analyse, procédez comme suit :

1. Dans l'arborescence XML de la sortie, cliquez-droit sur le nœud racine *customer (loop)* pour ouvrir le menu contextuel et sélectionnez **Delete**. Tous les éléments sous la racine *customers* sont supprimés. Vous pouvez reconstruire l'arborescence XML afin d'obtenir la meilleure structure pour grouper vos données.

2. Dans l'arborescence XML de la sortie, cliquez-droit sur le nœud racine *customers* afin d'ouvrir le menu contextuel et sélectionnez **Create Sub-Element**. Une boîte de dialogue s'ouvre.

3. Saisissez le nom du sous-élément. Dans ce scénario, le nom est *Customer*.

4. Répétez les opérations précédentes afin de créer deux sous-éléments supplémentaires sous le nœud *Customer*. Nommez-les : *LabelState* et *Name*.

5. Répétez ces opérations encore une fois afin de créer deux sous-éléments sous le nœud *Name*. Nommez-les : *id* et *CustomerName*.

6. Cliquez-droit sur le nœud *Name* pour ouvrir le menu contextuel et sélectionnez **As loop element** pour définir cet élément comme élément de boucle.

7. Cliquez-droit sur le nœud *Customer* pour ouvrir son menu contextuel et sélectionnez **As group element**. Cela signifie que les données de sortie sont regroupées dans l'élément *LabelState*.

8. Dans le flux Lookup de données, glissez-déposez la ligne *LabelState* dans la ligne du nœud *LabelState* de la colonne **Expression** dans la table de sortie. Les données correspondantes sont mappées.

9. Répétez l'opération pour mapper l'élément *id* et les éléments *CustomerName* des deux côtés.

10. Si nécessaire pour générer plusieurs flux XML, cliquez sur la clé anglaise en haut à droite afin d'ouvrir le panneau de paramétrage du mapping et configurer la fonctionnalité **All in one** comme **true**. Dans cet exemple, cette option est sur **true**. Pour plus d'informations concernant la fonctionnalité **All in one**, consultez le *Guide utilisateur du Studio Talend*.

11. Cliquez sur **OK** pour valider les modifications et fermer l'éditeur.

Si vous fermez le **Map Editor** sans avoir défini les éléments de boucle nécessaires, comme expliqué plus tôt dans ce scénario, l'élément racine est automatiquement défini comme élément de boucle.

12. Appuyez sur **F6** pour exécuter le Job.

    Les résultats d'exécution s'affichent comme suit dans la vue **Run**.

```
[statistics] connecting to socket on port 3517
[statistics] connected
<?xml version="1.0" encoding="UTF-8"?>
<customers><Customer><LabelState>Connecticut</LabelState><Name>
<id>1</id><CustomerName>Griffith Paving and
Sealcoatin</CustomerName></Name><Name><id>56</id><CustomerName>
Glenn Oaks Office
Supplies</CustomerName></Name></Customer><Customer><LabelState>
Ohio</LabelState><Name><id>2</id><CustomerName>Bill's Dive
Shop</CustomerName></Name><Name><id>61</id><CustomerName>DBN
Bank</CustomerName></Name></Customer><Customer><LabelState/><N
ame><id>62</id><CustomerName>BBQ Smith's Tex
Mex</CustomerName></Name></Customer><Customer><LabelState>Flor
ida</LabelState><Name><id>63</id><CustomerName>Pivot Point
College</CustomerName></Name></Customer></customers>
[statistics] disconnected
```

Les éléments *id* et *CustomerName* contenus dans la boucle sont regroupés dans l'élément *LabelState*. La balise du **group element** *Customer* marque le début et la fin de chaque groupe.

Le composant **tXMLMap** propose les éléments "**group**" et "**aggregate**" pour classer les données dans une structure XML. Lors de la gestion d'une ligne de données (flux XML complet), la différence de comportement est la suivante :

- L'élément "**group**" traite toujours les données dans un flux unique.
- L'élément "**aggregate**" sépare ce flux en différents flux XML complets.

Scénario 6 : Classer les données de sortie avec l'élément "aggregate"

A partir du Job dans *Scénario 5 : Mapper des données à l'aide d'un élément "group"*, ce scénario explique comment définir un élément "**aggregate element**" dans le **Map Editor** du **tXMLMap**, afin de classer les données de sortie dans des flux XML séparés.

Dans l'espace de modélisation graphique, double-cliquez sur le composant **tXMLMap** afin d'ouvrir son **Map editor**. La sortie s'affiche comme suit :

L'objectif de ce scénario est d'agréger les informations client selon les Etats de résidence, à l'aide de l'élément **aggregate**, puis d'envoyer ces données séparément dans différents flux XML vers le composant suivant.

Pour prendre en compte un élément **aggregate**, les données XML à traiter doivent avoir été triées, par exemple via vos outils XML, à l'aide de l'élément que vous souhaitez utiliser comme condition d'agrégation. La capture ci-dessous affiche une partie des données source triées. Les clients ayant le même *idState* sont regroupés.

```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<Customers>
    <Customer RegisterTime="2001-01-17 06:26:40.000">
        <Name>
            <id>1</id>
            <CustomerName>Griffith Paving and Sealcoatin</CustomerName>
        </Name>
        <Address>
            <CustomerAddress>talend@apres91</CustomerAddress>
            <idState>7</idState>
        </Address>
        <Revenue>
            <Suml>67852</Suml>
        </Revenue>
    </Customer>
    <Customer RegisterTime="1987-02-23 17:33:20.000">
        <Name>
            <id>56</id>
            <CustomerName>Glenn Oaks Office Supplies</CustomerName>
        </Name>
        <Address>
            <CustomerAddress>1859 Green Bay Rd.</CustomerAddress>
            <idState>7</idState>
        </Address>
        <Revenue>
            <Suml>1225.</Suml>
        </Revenue>
    </Customer>
    <Customer RegisterTime="2002-06-07 09:40:00.000">
        <Name>
            <id>2</id>
            <CustomerName>Bill's Dive Shop</CustomerName>
```


Pour ce faire, ajustez l'arborescence XML comme suit :

1. Cliquez-droit sur l'élément **Customer** afin d'ouvrir son menu contextuel et sélectionnez **Remove group element**.

2. Cliquez sur la clé anglaise en haut à droite afin d'ouvrir le panneau de paramétrage du mapping et configurer la fonctionnalité **All in one** comme **false**.

3. Cliquez-droit sur l'élément **LabelState** afin d'ouvrir son menu contextuel puis sélectionnez **As aggregate element**. Cet élément ajoute des balises relatives aux informations d'Etat de chaque client et les informations clients sont classées sous les informations d'Etats (State).

   Afin de rendre l'élément "**aggregate**" disponible, vérifiez que la fonctionnalité **All in one** est configurée sur **false**. Pour plus d'informations concernant cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.

4. Cliquez sur **OK** afin de valider ces modifications et fermer le **Map editor**.

5. Appuyez sur **F6** pour exécuter ce Job.

La vue **Run** s'ouvre automatiquement, dans laquelle vous pouvez vérifier les résultats d'exécution.

```
[statistics] connecting to socket on port 3526
[statistics] connected
<?xml version="1.0" encoding="UTF-8"?>
<customers><Customer><LabelState>Connecticut</LabelState><Name><id>1</id><Custo
merName>Griffith Paving and
Sealcoatin</CustomerName></Name><Name><id>56</id><CustomerName>Glenn Oaks
Office Supplies</CustomerName></Name></Customer></customers>
<?xml version="1.0" encoding="UTF-8"?>
<customers><Customer><LabelState>Ohio</LabelState><Name><id>2</id><CustomerName
Bill's Dive Shop</CustomerName></Name><Name><id>61</id><CustomerName>DBN
Bank</CustomerName></Name></Customer></customers>
<?xml version="1.0" encoding="UTF-8"?>
<customers><Customer><LabelState/><Name><id>62</id><CustomerName>BBQ Smith's
Tex Mex</CustomerName></Name></Customer></customers>
<?xml version="1.0" encoding="UTF-8"?>
<customers><Customer><LabelState>Florida</LabelState><Name><id>63</id><Customer
Name>Pivot Point College</CustomerName></Name></Customer></customers>
[statistics] disconnected
```

Le **tXMLMap** écrit quatre flux XML séparés, chacun d'entre eux contenant les informations d'un Etat et des clients vivant dans cet Etat.

Le composant **tXMLMap** propose les éléments "**group**" et "**aggregate**" pour classer les données dans une structure XML. Lors de la gestion d'une ligne de données (flux XML complet), la différence de comportement est la suivante :

- L'élément "**group**" traite toujours les données dans un flux unique.
- L'élément "**aggregate**" sépare ce flux en différents flux XML complets.

Scénario 7 : Restructurer des données produit à l'aide de différents éléments de boucle

Ce scénario utilise un Job à quatre composants pour restructurer les données produit fournies par un flux de type document, à l'aide de différents éléments de boucle (**loop element**).

Les composants utilisés sont :

- un **tFileInputXML** : lit les données produit source et les passe au **tXMLMap**.
- un **tXMLMap** : transforme le flux d'entrée pour lui donner la structure simplifiée attendue.
- un **tLogRow** : affiche les résultats d'exécution dans la console.
- un **tFileOutputXML** : génère un flux de sortie dans un fichier XML.

Le flux d'entrée se présente comme suit :

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<products category="1" name="laptop">

<!-- Summary -->
<summary>
 <company>DELL, HP</company>
 <sales unit="Dollars">12345678910.12345</sales>
 <model>business</model>
</summary>
```

```
<!-- Loop1 manufacture -->
<manufacture id="manu_1" date="2012-10-30">
 <name>DELL</name>
</manufacture>
<manufacture id="manu_2" date="2012-10-28">
 <name>HP</name>
</manufacture>

<!-- Loop2 types -->
<types model="business1">
 <type>DELL123</type>
 <manufacture_id>manu_1</manufacture_id>
</types>
<types model="business2">
 <type>HP123</type>
 <manufacture_id>manu_2</manufacture_id>
</types>

<!-- Loop3 sale -->
<sales>
 <sale unit="Dollars" type="DELL123">
  <quater>1</quater>
  <income>12345</income>
 </sale>
 <sale unit="Dollars" type="HP123">
  <quater>1</quater>
  <income>12345.123</income>
 </sale>
</sales>
</products>
```

L'objectif de cette restructuration est de simplifier la présentation des informations produit afin de réutiliser le fichier pour la manufacture.

Le flux de sortie est attendu comme suit :

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<root>
    <row>
        <outDoc>
            <manufactures category="1" name="laptop">
                <sales unit="Dollars">
                    <sale sales_type="DELL123">12345</sale>
                    <sale sales_type="HP123">12345.123</sale>
                </sales>
                <manufacture id="manu_1" date="2012-10-30" name="DELL"/>
                <manufacture id="manu_2" date="2012-10-28" name="HP"/>
                <types>
                    <type>DELL123</type>
                    <manufacture_id>manu_1</manufacture_id>
                </types>
                <types>
                    <type>DELL123</type>
                    <manufacture_id>manu_2</manufacture_id>
                </types>
                <types>
                    <type>HP123</type>
                    <manufacture_id>manu_1</manufacture_id>
                </types>
                <types>
                    <type>HP123</type>
                    <manufacture_id>manu_2</manufacture_id>
                </types>
            </manufactures>
        </outDoc>
    </row>
</root>
```

Dans le flux de sortie, l'élément racine est changé en *manufactures*, les informations commerciales sont sélectionnées et consolidées dans l'élément *sale* et l'élément *manufacture* est réduit à un seul niveau.

Pour reproduire ce scénario, procédez comme suit :

Déposer et relier les composants

Pour ce faire, effectuez les opérations suivantes :

1. Déposez un **tFileInputXML**, un **tXMLMap**, un **tLogRow** et un **tFileOutputXML** de la **Palette** dans l'espace de modélisation graphique.

2. Cliquez-droit sur le **tFileInputXML** afin d'ouvrir son menu contextuel et sélectionnez **Row** > **Main** afin de connecter ce composant au **tXMLMap**.

3. Répétez cette opération pour relier le **tXMLMap** au **tLogRow** à l'aide d'un lien **Row** > ***New Output* (Main)**. Une boîte de dialogue s'ouvre et vous demande de nommer ce lien de sortie. Dans ce scénario, nommez-le *outDoc*.

4. Répétez cette étape pour connecter le **tLogRow** au **tFileOutputXML** à l'aide d'un lien **Row** > **Main**.

Configurer le flux d'entrée

Pour ce faire, procédez comme suit :

1. Double-cliquez sur le **tFileInputXML** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir l'éditeur de schéma.

3. Cliquez sur le bouton **[+]** pour ajouter une ligne. Renommez-la *doc*.

4. Dans la colonne **Type**, sélectionnez **Document** dans la liste déroulante pour définir le type du flux d'entrée.

5. Dans le champ **File name/Stream**, parcourez votre système ou saisissez le chemin d'accès à la source XML fournissant les données d'entrée clients.

6. Dans le champ **Loop XPath query**, saisissez `"/"` pour remplacer la valeur par défaut. Cela signifie que les données source sont interrogées depuis la racine.

7. Dans la colonne **XPath query** de la table **Mapping**, saisissez la requête XPath. Dans ce scénario, saisissez `".."`, ce qui signifie que toutes les données de la source sont interrogées.

8. Dans la colonne **Get Nodes** de la table **Mapping**, cochez la case correspondante.

Configurer le tXMLMap avec différentes boucles

Pour ce faire, procédez comme suit :

1. Double-cliquez sur le composant **tXMLMap** pour ouvrir le **Map Editor**.

   Notez que la zone d'entrée est déjà renseignée avec la structure XML simple par défaut et que la table du haut est la table d'entrée principale.

2. Dans la table de gauche, cliquez-droit sur *doc* afin d'ouvrir le menu contextuel.

3. Dans ce menu contextuel, sélectionnez **Import From File** et, dans la boîte de dialogue qui s'ouvre, parcourez votre système jusqu'au fichier source correspondant, afin d'importer la structure XML utilisée par les données que le **tXMLMap** recevra. Dans ce scénario, le fichier source est *input.xml*, il fournit les données lues et chargées par le **tFileInputXML**.

4. Dans l'arborescence XML, cliquez-droit sur le nœud *manufacture* et sélectionnez **As loop element** afin de le sélectionner comme élément de boucle. Répétez l'opération avec les nœuds *types* et *sale*, respectivement.

5. Dans la partie inférieure de l'éditeur, cliquez sur l'onglet **Schema editor** pour afficher la vue correspondante.

6. Du côté droit de la vue, cliquez sur le bouton **[+]** pour ajouter une ligne à la table *outDoc*. Renommez la ligne *outDoc*.

7. Dans la colonne **Type** de la ligne *outDoc*, sélectionnez le type de données **Document**. La racine XML correspondante est ajoutée automatiquement à la table en haut à droite représentant le flux de sortie.

8. Dans la table *outDoc*, importez la structure XML des données que vous devez utiliser depuis le fichier source XML correspondant. Dans ce scénario, le fichier est *ref.xml*. Ce fichier fournit la structure XML attendue.

9. Cliquez-droit sur le nœud *manufacture* et sélectionnez **As loop element** dans le menu contextuel. Répétez l'opération afin de sélectionner les nœuds *types* et *sale* comme éléments de boucle.

   Vous pouvez commencer à mapper le flux d'entrée vers le flux de sortie.

10. Dans la table en haut à gauche de l'éditeur, cliquez sur le nœud *@category* et déposez-le dans la colonne **Expression**, dans la ligne de sortie correspondante. Dans ce scénario, c'est le nœud *@category*.

11. Répétez l'opération afin de mapper :

- *@name* à *@name*
- *@unit* sous le nœud *summary* à *@unit*
- *@id* à *@id* et à *manufacture id*, respectivement
- *@date* à *@date*
- *name* à *@name*
- *type* à *type*
- *@type* à *@sales_type*
- *income* à *sale (loop)*

12. Si vous ne devez générer qu'un seul flux XML, cliquez sur l'icône de clé anglaise en haut de la table de sortie afin d'ouvrir le panneau de paramétrage et configurez la fonctionnalité **All in one** à **true**. Pour plus d'informations concernant la fonctionnalité **All in one**, consultez le *Guide utilisateur du Studio Talend*.

13. Cliquez sur le bouton **[...]** à côté de l'élément de boucle **manufacture** afin d'ouvrir la boîte de dialogue **[Configure source loops]** puis cliquez une fois sur le bouton **[+]** pour ajouter une boucle source, *manufacture*. Répétez l'opération pour l'élément de boucle **sale** et ajoutez une boucle source, *sale*.

14. Cliquez sur le bouton **[...]** à côté de l'élément boucle **types** pour ouvrir la boîte de dialogue **[Configure source loop]**. Cliquez deux fois sur le bouton **[+]** pour ajouter deux boucles source, *types* et *manufacture*. Assurez-vous que le numéro de séquence de la boucle source *types* est *0*. Ainsi, la partie relative du flux de sortie sera triée selon les valeurs de l'élément *type*.

Lorsqu'un élément de boucle reçoit des mappings d'un ou plusieurs élément(s) de boucle du flux d'entrée, cela vous permet de configurer la séquence des boucles en entrée. Par exemple, dans ce scénario, l'élément de boucle *types* du flux de sortie est mappé avec *@id*, appartenant à l'élément de boucle *manufacture* et *type*, appartenant à l'élément de boucle *types*. Le flux de sortie est trié selon la boucle primaire *types*.

15. Cliquez sur **OK** pour valider les mappings et fermer le **Map Editor**.

Si vous fermez le **Map Editor** sans avoir configuré les éléments de boucle requis, l'élément racine sera automatiquement défini comme élément de boucle.

Configurer le flux de sortie

Pour ce faire, procédez comme suit :

1. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Component**.

2. Si ce composant n'a pas le même schéma que le composant le précédant, une icône d'avertissement apparaît. Dans ce cas, cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Cela fait, l'icône d'avertissement disparaît.

3. Cliquez sur **OK** afin de valider ces modifications et acceptez la propagation proposée par la boîte de dialogue qui s'ouvre.

4. Double-cliquez sur le **tFileOutputXML** pour ouvrir sa vue **Component**.

5. Dans le champ **File Name**, parcourez votre système ou saisissez le chemin d'accès au fichier dans lequel générer le flux de sortie.

Exécuter le Job

Pour exécuter ce Job, appuyez sur **F6**.

La vue **Run** s'ouvre automatiquement. Vous pouvez vérifier les résultats d'exécution.

```
[statistics] connecting to socket on port 3514
[statistics] connected
<?xml version="1.0" encoding="UTF-8"?>
<manufactures category="1" name="laptop"><sales unit="Dollars"><sale
sales_type="DELL123">12345</sale><sale
sales_type="HP123">12345.123</sale></sales><manufacture id="manu_1"
date="2012-10-30" name="DELL"/><manufacture id="manu_2" date="2012-10-28"
name="HP"/><types><type>DELL123</type><manufacture_id>manu_1</manufacture_id></t
ypes><types><type>DELL123</type><manufacture_id>manu_2</manufacture_id></types><
types><type>HP123</type><manufacture_id>manu_1</manufacture_id></types><types><t
ype>HP123</type><manufacture_id>manu_2</manufacture_id></types></manufactures>
[statistics] disconnected
```

Ouvrez le fichier généré. Vous pouvez voir les données produit attendues restructurées pour la manufacture.

Composants System

Ce chapitre passe en revue les principaux composants de la famille **System** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **System** regroupe les composants permettant d'interagir avec votre système d'exploitation.

tRunJob

Propriétés du tRunJob

Famille de composant	System	
Fonction	Le composant **tRunJob** exécute le Job spécifié dans ses propriétés dans un contexte défini.	
Objectif	Le **tRunJob** permet de maîtriser des systèmes de Job complexes devant être exécuter l'un après l'autre.	
Basic settings	*Schema* et *Edit Schema* *(Non disponible dans un Job Map/Reduce)*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Copy Child Job Schema *(Non disponible dans un Job Map/Reduce)*	Cliquez sur ce bouton pour récupérer le schéma du Job fils.
	Use dynamic job	Cochez cette case afin de pouvoir appeler et traiter de nombreux Jobs. Lorsque cette case est cochée, seule la dernière version des Jobs peut être appelée et traitée. Un processus indépendant sera utilisé pour exécuter le sous-job. Les options **Context** et **Use an independent process to run subjob** disparaissent. *L'option **Use dynamic job** n'est pas compatible avec le cache du JobServer. Dès lors, l'exécution peut échouer si vous exécutez dans un Job contenant un composant **tRunJob** avec cette option activée.* *Cette option est incompatible avec l'option **Use or register a shared DB Connection** des composants de base de données. Quand le **tRunJob** fonctionne de paire avec un composant de connexion à une base de données, activer les deux options fera échouer l'exécution du Job.*
	Context job	Ce champ est visible uniquement lorsque l'option **Use dynamic job** est cochée. Saisissez le nom du Job que vous souhaitez appeler dans la liste des Jobs sélectionnés.

	Job	Sélectionnez le Job à appeler et à traiter. Assurez-vous que le Job appelé ait été exécuté au moins une fois auparavant pour que son exécution via le **tRunJob** se déroule le mieux possible.
	Version	Sélectionnez la version du Job fils que vous souhaitez utiliser.
	Context	Si vous avez paramétré des contextes et des variables pour les Jobs à exécuter via le **tRunJob**, sélectionnez dans la liste le contexte à appliquer.
	Use an independent process to run subjob *(Non disponible dans un Job Map/Reduce)*	Cochez cette case afin d'utiliser un processus indépendant pour exécuter le sous-job. Cela vous permettra de résoudre des problèmes liés à la taille limite de la mémoire. *L'option **Use dynamic job** n'est pas compatible avec le cache du JobServer. Dès lors, l'exécution peut échouer si vous exécutez dans un Job contenant un composant **tRunJob** avec cette option activée.* *Cette option n'est pas compatible avec l'option **Use or register a shared DB Connection** des composants de connexion à une base de données. Lorsque le **tRunJob** fonctionne avec un composant de connexion à une base de données, activer les deux options peut faire échouer votre Job.*
	Die on child error	Décochez cette case pour exécuter le Job parent même s'il y a une erreur lors de l'exécution du Job fils.
	Transmit whole context	Cochez cette case pour obtenir les contextes du Job parent. Décochez-la pour obtenir les contextes du Job fils. Si cette case est cochée lorsque les Jobs père et fils ont les mêmes variables de contexte définies : • les valeurs des variables pour le Job père sont utilisées durant l'exécution du Job fils si aucune valeur cohérente n'est définie dans la table **Context Param**. • sinon, les valeurs définies dans la table **Context Param** sont utilisées durant l'exécution du Job fils.
	Context Param	Vous pouvez modifier la valeur du contexte sélectionné. Cliquez sur le bouton **[+]** pour ajouter les paramètres définis dans l'onglet **Context** contexte du Job fils. Pour plus d'informations sur les paramètres de contexte, consultez le *Guide utilisateur du Studio Talend*. Les valeurs définies ici sont utilisées durant l'exécution du Job fils, même si la case **Transmit whole context** est cochée.
Advanced settings	*Propagate the child result to the output schema* *(Non disponible dans un Job Map/Reduce)*	Cochez cette case afin de propager les données de sortie stockées dans la mémoire tampon via le composant **tBufferOutput** du Job fils vers le composant de sortie du Job père. Cette case est décochée par défaut. Elle ne s'affiche pas lorsque vous cochez la case **Use dynamic job** ou **Use an independent process to run subjob**.
	Print Parameters	Cochez cette case pour afficher les paramètres internes et externes dans la **Console**.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **CHILD_RETURN_CODE** : code de retour d'un Job fils. Cette variable est une variable **After** et retourne un nombre entier. **CHILD_EXCEPTION_STACKTRACE** : trace de la pile d'un Job fils. Cette variable est une variable **After** et retourne une chaîne de caractères.	

	Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError, Synchronize, Parallelize. Dans un Job Map/Reduce, seuls les liens **Trigger** sont disponibles pour ce composant. Pour plus d'informations concernant les liens, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en Job standalone ou peut aider à clarifier certains Jobs complexes en évitant d'avoir un nombre important de sous-jobs dans le même Job.
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez également utiliser ce composant en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme étape intermédiaire ou finale. Il génère du code Map/Reduce natif qui peut être exécuté directement dans Hadoop. Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	L'option **Use dynamic job** du composant **tRunJob** permet au Job principal d'appeler le Job fils dynamiquement, lors de l'exécution. Cependant, cette option ne fonctionne pas si le Job fils a été exporté avec un type **Axis WebService (WAR)** et déployé dans Tomcat en tant que service Web. Pour plus d'informations concernant une solution alternative, consultez la page https://help.talend.com/pages/viewpage.action?pageId=27029205 (en anglais).

Scénario 1 : Exécuter un Job fils

Le scénario suivant décrit un Job à un composant appelant et exécutant un autre Job. Le Job à exécuter lit une liste de fichiers délimités simples et affiche leur contenu dans la console de la vue **Run**.

Créer le Job fils

Déposer et relier les composants

1. A partir de la **Palette**, cliquez et déposez les composants **tFileInputDelimited** et **tLogRow** dans l'espace de modélisation graphique.

2. Connectez ces deux composants à l'aide d'un lien de type **Row Main**.

Configurer les composants

1. Double-cliquez sur le composant **tFileInputDelimited** pour afficher la vue **Basic settings** et paramétrer les propriétés du composant d'entrée.

2. Cliquez dans le champ **File Name** puis appuyez sur la touche **F5** pour ouvrir la fenêtre de configuration de variable.

3. Dans le champ **Name**, saisissez un nom pour cette nouvelle variable de contexte. Nommez-la *File*, dans cet exemple.
4. Dans le champ **Defaut value**, saisissez le chemin d'accès au fichier d'entrée par défaut.
5. Cliquez sur **Finish** pour valider puis appuyez sur **Entrée** pour vous assurer que la nouvelle variable de contexte est conservée dans le champ **File Name**.

   Vous pouvez également créer ou éditer un paramètre de contexte dans l'onglet **Context** au bas de l'espace de modélisation graphique. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*

6. Cliquez sur le bouton **[...]** près de **Edit schema** pour ouvrir la boîte de dialogue **[Schema]**, dans laquelle vous pouvez configurer le schéma manuellement.
7. Dans la boîte de dialogue, cliquez sur le bouton **[+]** pour ajouter des colonnes et nommez-les en fonction de la structure du fichier d'entrée.

   Dans cet exemple, ce composant lit les fichiers définis dans le Job parent. Ces fichiers contiennent jusqu'à cinq colonnes. Ajoutez cinq colonnes de type **String** et appelez-les respectivement *col_1*, *col_2*, *col_3*, *col_4* et *col_5*. Cliquez ensuite sur **OK** pour valider la configuration du schéma et fermer la boîte de dialogue **[Schema]**.
8. Double-cliquez sur le composant **tLogRow** pour afficher sa vue **Basic settings** et paramétrer ses propriétés.
9. Sélectionnez l'option **Table** pour visualiser le contenu dans des cellules de tableau.

Créer le Job parent

Déposer et relier les composants

1. Cliquez et déposez les composants **tFileList** et **tRunJob** dans l'espace de modélisation.

2. Connectez ces deux composants à l'aide d'un lien de type **Iterate**.

Configurer les composants

1. Double-cliquez sur le composant **tFileList** pour afficher la vue **Basic settings** et paramétrer les propriétés du composant.

2. Dans le champ **Directory**, renseignez le chemin d'accès au répertoire contenant les fichiers à traiter ou cliquez sur le bouton **[...]** pour parcourir votre système jusqu'à ce répertoire. Dans cet exemple, il s'agit du répertoire *tRunJob* qui contient trois fichiers délimités.

3. Dans la liste déroulante **FileList Type**, sélectionnez l'option **Files**.

4. Cochez la case **Use Glob Expressions as Filemask** pour pouvoir utiliser des expressions régulières dans vos masques de fichier et cliquez sur le bouton **[+]** pour ajouter une ligne dans la zone **Files** et définir un filtre pour faire correspondre les fichiers. Dans cet exemple, saisissez `.csv` pour récupérer tous les fichiers délimités.

5. Double-cliquez sur le composant **tRunJob** pour afficher la vue **Basic settings** et paramétrer les propriétés du composant.

6. Cliquez sur le bouton **[...]** à côté du champ **Job** pour accéder à la boîte de dialogue **[Find a Job]**.

7. Sélectionnez le Job fils à exécuter puis cliquez sur **OK** pour fermer la boîte de dialogue. Le nom du Job sélectionné apparaît dans le champ **Job** de la vue **Basic settings** du **tRunJob**.

8. Dans la zone **Context Param**, cliquez sur le bouton **[+]** pour ajouter un paramètre de contexte. Le seul paramètre de contexte défini dans le Job fils, *File*, apparaît dans la cellule **Parameter**.

9. Cliquez dans la cellule **Values** puis appuyez sur **Ctrl+Espace** pour accéder à la liste des variables de contexte et sélectionnez le contexte *tFileList-1.CURRENT_FILEPATH*.

   La variable de contexte correspondante apparaît dans la cellule **Values** : `((String)globalMap.get(tFileList-1.CURRENT_FILEPATH))`

   Pour plus d'informations concernant les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

Exécuter le Job parent

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** pour exécuter votre Job.

```
.---------------------+-----+-----+-----+-----.
|                   tLogRow_1                   |
|=--------------------+-----+-----+-----+----=|
|col_1                |col_2|col_3|col_4|col_5|
|=--------------------+-----+-----+-----+----=|
|email                |     |     |     |     |
|mmoore@hotmail.com   |     |     |     |     |
|ataylor@gmail.com4   |     |     |     |     |
|canderson@foxmail.com|     |     |     |     |
'---------------------+-----+-----+-----+-----'

.-----+----------+---------+---------+------.
|                 tLogRow_1                  |
|=----+----------+---------+---------+-----=|
|col_1|col_2     |col_3    |col_4    |col_5 |
|=----+----------+---------+---------+-----=|
|id   |firstName |lastName |company  |city  |
|1    |Michael   |Jackson  |IBM      |Roma  |
|2    |Elisa     |Black    |Microsoft|London|
```

Le Job appelé lit les données contenues dans les fichiers d'entrée comme défini dans le schéma d'entrée et le résultat de ce Job est directement affiché dans la console **Run**.

Voir également :

- *tLoop*,
- *Scénario 1 : Mettre des données en mémoire tampon* du composant **tBufferOutput**.

Scénario 2 : Exécuter une liste de Jobs fils de manière dynamique

Ce scénario décrit un Job qui appelle deux Jobs fils de manière dynamique. Lorsqu'ils sont appelés par le Job père, chacun des Jobs fils affiche un message dans la console.

Configurer les Jobs fils

1. Créez un Job nommé *child_1* et ajoutez deux composants en tapant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette** :
   - un **tFixedFlowInput**, afin de générer un message et
   - un **tLogRow**, afin d'afficher le message généré dans la console.
2. Reliez le **tFixedFlowInput** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

3. Double-cliquez sur le **tFixedFlowInput** afin d'ouvrir sa vue **Basic settings**.

4. Cliquez sur le bouton **[...]** à côté du bouton **Edit schema** afin d'ouvrir la boîte de dialogue **[Schema]** et définir la schéma d'entrée.

   Dans cet exemple, le schéma a une seule colonne : *Message* (de type **String**).

   Une fois le schéma défini, cliquez sur **OK** afin de fermer l'éditeur. Dans la boîte de dialogue qui s'affiche, cliquez sur **Yes** afin de propager les changements au composant suivant.

5. Cochez la case **Use Inline Content** et, dans le champ **Content**, saisissez le message que vous souhaitez afficher dans la console, *Hello World!* dans cet exemple.

6. Dans la vue **Basic settings** du **tLogRow**, sélectionnez le mode **Table** afin d'afficher le résultat de l'exécution sous forme de tableau.

7. Créez une copie de ce Job et nommez-le *child_2*. Dans le champ **Content** du **tFixedFlowInput**, saisissez un autre message, *Hello Talend!* dans cet exemple.

8. Appuyez sur **Ctrl+S** afin de sauvegarder les Jobs.

Configurer le Job père

Ajouter et relier les composants

1. Créez un nouveau Job et ajoutez les composants suivants en tapant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette** :

   - un **tFixedFlowInput**, afin de définir dynamiquement la liste des Jobs à appeler,
   - un **tFlowToIterate**, afin d'effectuer une itération sur les lignes en entrée et stocker le contenu dans une variable globale d'itération et
   - un **tRunJob**, afin de charger et exécuter les Jobs fils.

2. Reliez le **tFixedFlowInput** au **tFlowToIterate** à l'aide d'un lien **Row** > **Main**.

3. Reliez le **tFlowToIterate** au **tRunJob** à l'aide d'un lien **Row** > **Iterate**.

Configurer les composants

1. Double-cliquez sur le **tFixedFlowInput** afin d'ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir la boîte de dialogue **[Schema]** et définir le schéma d'entrée.

Dans cet exemple, le schéma a une seule colonne : *Job* (de type **String**).

Une fois le schéma défini, cliquez sur **OK** afin de fermer l'éditeur. Dans la boîte de dialogue qui s'affiche, cliquez sur **Yes** afin de propager les changements au composant suivant.

3. Cochez la case **Use Inline Content** et, dans le champ **Content**, spécifiez le nom des Jobs fils qui sont appelés par le Job père :

```
child_1
child_2
```

4. Double-cliquez sur le **tRunJob** afin d'ouvrir sa vue **Basic settings**.

5. Cochez la case **Use dynamic job**.

6. Cliquez dans le champ **Context job**, appuyez sur **Ctrl+Espace** et, dans la liste de variable, sélectionnez la variable d'itération créée par le composant **tFlowToIterate**, *Job* dans cet exemple. Le champ **Context job** contient à présent `((String)globalMap.get("row1.Job"))`. A chaque itération, cette variable est remplacée par le nom du Job appelé.

7. Cliquez sur le bouton **[...]** à côté du champ **Job** afin d'ouvrir la boîte de dialogue **[Select Job]**. Sélectionnez tous les Jobs que vous souhaitez exécuter et cliquez sur **OK** afin de fermer la boîte de dialogue.

1. Appuyez sur **Ctrl+S** afin de sauvegarder le Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la console **Run** afin d'exécuter le Job.

Les Jobs fils sont appelés l'un après l'autre et les messages spécifiés dans les Jobs fils sont affichés dans la console.

Scénario 3 : Propager les données de sortie en mémoire tampon du Job fils au Job père

Dans ce scénario, un Job à trois composants appelle un Job fils de deux composants et affiche dans la console les données de sortie du Job fils mises en mémoire tampon, au lieu des données du flux d'entrée du Job père.

Construire le Jobs fils

1. Créez un Job nommé *child* et ajoutez deux composants en cliquant puis saisissant leur nom dans l'espace de modélisation graphique ou en les glissant de la **Palette** dans l'espace de modélisation graphique :
   - un **tFixedFlowInput**, pour générer un message,
   - un **tBufferOutput**, pour stocker le message généré dans la mémoire tampon.
2. Reliez le composant **tFixedFlowInput** au **tBufferOutput** à l'aide d'un lien **Row** > **Main**.
3. Double-cliquez sur le composant **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

4. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir la boîte de dialogue **[Schema]** et définir le schéma des données d'entrée. Dans cet exemple, le schéma contient une seule colonne *message* de type String.

   Cela fait, cliquez sur **OK** pour valider les modifications puis sur **Yes** dans la boîte de dialogue **[Propagate]** afin de propager le schéma au composant suivant.

5. Dans la zone **Mode**, sélectionnez l'option **Use Single Table** et définissez la valeur de la colonne *message* dans la table **Values**. Dans cet exemple, la valeur est *"message from the child job"*.

Construire le Job père

1. Créez un Job, ajoutez trois composants en cliquant et saisissant leur nom dans l'espace de modélisation graphique ou en les déposant de la **Palette** dans l'espace de modélisation graphique :
   - un **tFixedFlowInput**, pour générer un message,
   - un **tRunJob**, pour appeler le Job nommé *child*
   - un **tLogRow**, pour afficher les résultats d'exécution dans la console.
2. Connectez le composant **tFixedFlowInput** au **tRunJob** et le **tRunJob** au composant **tLogRow** à l'aide d'un lien **Row** > **Main**.
3. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

4. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir la boîte de dialogue **[Schema]** et définir le schéma des données d'entrée. Dans cet exemple, le schéma contient une seule colonne *message*, de type String.

Cela fait, cliquez sur **OK** pour valider les modifications.

5. Dans la zone **Mode**, sélectionnez l'option **Use Single Table** et définissez la valeur de la colonne *message* de la table **Values**. Dans cet exemple, la valeur est *"message from the parent job"*.

6. Cliquez sur le **tRunJob** puis cliquez sur l'onglet **Component** pour ouvrir sa vue **Basic settings**.

7. Cliquez sur le bouton **Sync columns** puis sur **Yes** dans la boîte de dialogue **[Propagate]** afin de récupérer le schéma du composant précédent.

8. Cliquez sur le bouton **[...]** à côté du champ **Job** pour ouvrir la boîte de dialogue **[Repository Content]**.

Dans la boîte de dialogue **[Repository Content]**, sélectionnez le Job nommé *child* puis cliquez sur **OK** pour fermer la boîte de dialogue.

9. Dans la vue **Advanced settings** du composant **tRunJob**, cochez la case **Propagate the child result to the output schema**. Cette case cochée, la sortie en mémoire tampon du Job fils will be propagated to the output component.

Exécuter le Job père

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** afin d'exécuter le Job.

Le Job fils est appelé et le message défini dans le Job fils s'affiche dans la console, à la place du message défini dans le Job père.

tSetEnv

Propriétés du tSetEnv

Famille du composant	System	
Fonction	Le composant **tSetEnv** ajoute des variables à l'environnement du système de manière temporaire, pendant l'exécution du Job.	
Objectif	Le **tSetEnv** permet de créer des variables et d'exécuter un Job script en communicant les informations des variables nouvellement créées entre les différents sous-Jobs. Après l'exécution du Job, ces nouvelles variables sont supprimées.	
Basic settings	*Parameters*	**Cliquez sur le bouton [+] pour ajouter les variables nécessaires au Job. name** : Renseignez la syntaxe de la nouvelle variable. **value** : Entrez une valeur pour ce paramètre en fonction du contexte. **append** : Cochez cette case pour ajouter la nouvelle variable à la fin.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Le **tSetEnv** peut être utilisé comme composant de début ou composant intermédiaire.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario : Modifier une variable lors de l'exécution d'un Job

Le scénario suivant se présente sous la forme de deux Jobs père et fils. Il permet grâce au **tSetEnv** de transférer et de modifier dans un Job fils une valeur créée dans un Job parent. Dans le cadre de ce Job, les composants **tMsgBox** sont utilisés pour afficher à titre informatif qu'une variable est correctement appliquée, via une fenêtre pop-up.

Pour modifier la valeur du Job père en utilisant une variable créée par le **tSetEnv** dans le Job fils, procédez comme décrit dans les sections suivantes :

Déposer et relier les composants

1. Créez un premier Job appelé *parentJob* : cliquez-droit sur l'onglet **Job Design** du **Repository**, puis sélectionnez **Create Job**.

2. A partir de la **Palette**, glissez-déposez un composant **tSetEnv**, deux composants **tMsgBox** et un **tRunJob** dans l'espace de modélisation graphique.

3. Connectez le **tSetEnv** à un premier **tMsgBox** à l'aide d'un lien **OnSubjobOk** : cliquez-droit sur le composant de départ, sélectionnez *Trigger*, puis **OnSubjobOk**, puis cliquez sur le composant à relier.

4. Connectez le premier **tMsgBox** au **tRunJob** également à l'aide d'un lien **OnSubjobOk**.

5. Connectez ensuite le **tRunJob** au second **tMsgBox** également à l'aide d'un lien **OnSubjobOk**.

6. Créez maintenant le Job fils, appelé *ChildJob*.

7. A partir de la **Palette**, glissez-déposez un **tSetEnv** et un**tMsgBox**.

8. Connectez le **tSetEnv** au **tMsgBox** à l'aide d'un lien **OnSubjobOk** : cliquez-droit sur le composant de départ, sélectionnez *Trigger*, puis **OnSubjobOk**. Puis cliquez sur le composant à relier.

Configurer les composants

Dans cet exemple, la valeur définie dans le Job parent est transférée au Job fils. Elle y est modifiée pour prendre la valeur du Job fils et retransférée vers le Job parent.

1. Dans le Job *ParentJob*, sélectionnez le **tSetEnv** puis cliquez sur l'onglet **Component**. Ajoutez une ligne de variable en cliquant sur le bouton **[+]** pour définir sa valeur initiale. Saisissez *Variable_1* dans le champ **Name** et *Parent job value* dans le champ **Value**.

2. Sélectionnez le premier **tMsgBox** et cliquez sur l'onglet **Component**. Dans le champ **Message**, saisissez le message qui s'affiche dans votre fenêtre pop-up et qui confirme que votre variable est bien prise en compte dans le Job. Par exemple : `"Parent:"+System.getProperty("Variable_1")` affiche la variable définie dans le **tSetEnv** (ici *Parent job value*).

3. Sélectionnez le second **tMsgBox** et cliquez sur l'onglet **Component**. Dans le champ **Message**, saisissez à nouveau la ligne `"Parent:"+System.getProperty("Variable_1")`. Elle fait apparaître dans le Job parent la variable définie dans le Job fils.

4. Sélectionnez le **tRunJob** et cliquez sur l'onglet **Component**. Dans le champ **Job**, saisissez le nom de votre Job fils, ici *ChildJob*. Cela exécute le Job fils à l'exécution du Job parent.

5. Double-cliquez sur le **tRunJob** pour ouvrir le Job fils *ChildJob*.

6. Sélectionnez le composant **tSetEnv** et cliquez sur l'onglet **Component**. Ajoutez une ligne de variable en cliquant sur le bouton **[+]** pour définir sa valeur initiale. Saisissez *Variable_1* dans le champ **Name** et *Child job value* dans le champ **Value**.

7. Sélectionnez ensuite le **tMsgBox** et cliquez sur l'onglet **Component**. Dans le champ **Message**, saisissez le message qui s'affiche dans votre fenêtre pop-up et qui confirme que votre variable est correctement prise en compte dans le Job. Par exemple, la ligne de code `"Son:"+System.getProperty("Variable_1")` affiche la variable définie dans le **tSetEnv** (ici *Child job value*).

8. Enregistrez votre Job, retournez sous le Job parent, puis appuyez sur **F6** pour l'exécuter.

Exécuter le Job

Trois fenêtres pop-up s'affichent l'une après l'autre :

- *Parent: Parent job value* : la valeur du Job parent est *Parent job value*.
- *Child: Child job value* : la valeur du Job fils est *Child job value*.
- *Parent: Child job value* : la valeur du Job parent a été modifiée par la variable définie dans le **tSetEnv** du Job fils, puis retransférée au Job parent. La valeur du Job parent est dorénavant celle définie dans le Job fils : *Child job value*.

tSSH

Propriétés du tSSH

Famille de composant	System	
Fonction	Le composant **tSSH** renvoie les données d'un ordinateur distant via le protocole Secure Shell défini.	
Objectif	Le **tSSH** permet d'entrer en communication avec un serveur distant et de renvoyer les informations de sécurité.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Host	Adresse IP.
	Port	Numéro du port d'écoute.
	User	Informations d'authentification de l'utilisateur.
Authentication method	*Public Key*/Key *Passphrase*/ *Private Key*	Sélectionnez l'option adéquate. Si vous choisissez l'option **Public Key**, saisissez la passphrase (phrase secrète), si nécessaire, dans le champ **Key passphrase**. Puis, dans le champ **Private Key**, saisissez la clé privée ou cliquez sur le bouton **[...]** à côté du champ **Private Key** pour parcourir votre répertoire jusqu'à la clé privée. Pour saisir la phrase secrète, cliquez sur le bouton **[...]** à côté du champ **Passphrase**, puis, dans la boîte de dialogue qui s'ouvre, saisissez la phrase secrète entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
Authentication method	*Password/Password*	Sélectionnez l'option adéquate.

		Si vous choisissez l'option **Password**, saisissez votre mot de passe dans le champ **Password**. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
Authentication method	*Keyboard Interactive/ Password*	Sélectionnez l'option adéquate. Si vous choisissez l'option **Keyboard Interactive**, saisissez le mot de passe requis dans le champ **Password**. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Pseudo terminal	Cochez cette case pour appeler un interpréteur de commandes interactif effectuant les opérations de terminal.
	Command separator	Saisissez le séparateur de commandes requis. Lorsque la case **Pseudo terminal** est cochée, ce champ n'est plus disponible.
	Commands	Saisissez la commande permettant de récupérer l'information de l'ordinateur distant. Lorsque la case **Pseudo terminal** est cochée, ce tableau devient un émulateur de terminal, et chaque ligne du tableau devient une commande.
	Use timeout/timeout in seconds	Définit la durée avant expiration. Un message notifiant l'expiration sera généré si l'actuel temps de réponse dépasse ce délai d'expiration.
	Standard Output	Sélectionnez la destination de la sortie standard à retourner La sortie peut être retournée vers : - la console (**to console**) : la sortie s'affiche dans la console de la vue **Run**. - les variables globales (**to global variables**) : la sortie est indiquée par la variable globale correspondante. - la console et les variables globales (**both to console and global variable**) : la sortie est indiquée par les deux moyens. - une sortie normale (**normal**) : la sortie est une sortie SSH standard.
	Error Output	Sélectionnez la destination de la sortie d'erreurs à retourner. La sortie peut être retournée vers : - la console (**to console**) : la sortie s'affiche dans la console de la vue **Run**. - les variables globales (**to global variables**) : la sortie est indiquée par la variable globale correspondante. - la console et les variables globales (**both to console and global variable**) : la sortie est indiquée par les deux moyens. - une sortie normale (**normal**) : la sortie est une sortie SSH standard.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : Run if, On Component Ok, On Component Error, On Subjob Ok, On Subjob Error. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Iterate.	

	Trigger : Run if, On Component Ok, On Component Error, On Subjob Ok, On Subjob Error. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **STDOUT** : sortie de l'exécution standard de la commande distante. Cette variable est une variable **After** et retourne un nombre entier. **STDERR** : sortie de l'exécution en erreur de la commande distante. Cette variable est une variable **After** et retourne un nombre entier. **EXIT_CODE** : code de sortie de la commande distante. Cette variable est une variable **After** et retourne un nombre entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant peut être utilisé en standalone. Pour les informations sensibles, telles que le mot de passe et l'identifiant, vous pouvez définir les variables de contexte dans l'onglet **Contexts** et appuyer sur les touches **Ctrl+Espace** pour y accéder.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
Limitation	L'utilisation de ce composant est adaptée pour les systèmes de type Unix.

Scénario : Accéder aux informations du système distant via SSH

Le scénario suivant décrit un Job utilisant un protocole SSH pour afficher le nom d'hôte du serveur distant auquel vous souhaitez vous connecter et la date actuelle de ce système distant.

Un composant **tSSH** est suffisant pour ce Job. Cliquez et déposez-le de la famille **System** de la **Palette** dans l'espace de modélisation graphique.

Double-cliquez sur le **tSSH** et sélectionnez l'onglet **Basic settings** de la vue **Component**.

1. Renseignez le nom de l'hôte dans le champ **Host** auquel accéder à partir du SSH, ainsi que le numéro du port dans le champ **Port**.

2. Dans le champ **User**, renseignez le nom d'identification de l'utilisateur sur l'ordinateur distant.

3. Sélectionnez la méthode d'authentification dans la liste déroulante **Authentication method**. Dans cet exemple, la méthode d'authentification utilisée est **Public key** (clé publique).

4. Renseignez donc le champ **Private key** avec votre clé privée.

5. Dans le champ **Command**, saisissez la commande suivante. Dans cet exemple, saisissez `hostname; date` entre guillemets doubles (code Java).

6. Cochez la case **Use timeout** et paramétrez la durée avant expiration à 5 secondes.

```
Starting job uniteElisa at 16:26 26/09/2007.
picasso
Wed Sep 26 14:24:15 CEST 2007
Job uniteElisa ended at 16:26 26/09/2007. [exit code=0]
```

L'ordinateur distant renvoie le nom de l'hôte, ainsi que la date et l'heure en cours.

tSystem

Propriétés du tSystem

Famille de composant	System	
Fonction	Le composant **tSystem** exécute une ou plusieurs commande(s) système.	
Objectif	**tSystem** peut appeler d'autres commandes de traitement, fonctionnant déjà dans un Job plus grand.	
Basic settings	*Use home directory*	Cochez cette case pour changer le nom et le chemin du répertoire dédié.
	Use Single Command	Lorsque la commande requise est particulièrement simple, par exemple si un paramètre est utilisé, sans espace, sélectionnez cette option dans le champ **Command**. Dans ce champ, saisissez la commande système simple à exécuter. Notez que la syntaxe n'est pas vérifiée. *Sous Windows, les commandes MS-DOS ne permettent pas de passer directement du dossier courant au dossier contenant le fichier à exécuter. Pour exécuter un fichier, vous devez donc utiliser une première commande afin de changer de dossier courant, puis une deuxième commande exécutant le fichier.*

	Use Array Command	Sélectionnez cette option afin d'activer son champ **Command**. Dans ce champ, saisissez la commande système, un paramètre par ligne.
		
		Par exemple, saisissez la commande suivante avec les espaces consécutifs, pour Linux :
		
		`"cp"`
		`"/temp/source.txt"`
		`"/temp/copy  to/"`

	Standard Output et *Error Output*	Sélectionnez le type de sortie auquel les données traitées seront transférées.
		to console : transmet les données vers la console de la vue **Run**.
		to global variable : les données sont envoyées dans une variable de sortie liée au composant **tSystem**.
		both to console and to global variable : les données sont envoyées à la fois vers la console de la vue **Run** et dans une variable de sortie liée au composant **tSystem**.
		normal : les données sont envoyées au composant suivant.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

		• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Environment variables	Cliquez sur le bouton **[+]** pour ajouter les variables d'environnement nécessaires au Job. **name** : Renseignez la syntaxe de la nouvelle variable. **value** : Entrez une valeur pour ce paramètre en fonction du contexte.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Utilisation	Ce composant sert aux entreprises qui utilisent déjà d'autres applications qu'ils souhaitent intégrer à leur flux via Talend.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : OnSubjobOk, OnSubjobError, Run if. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main, Reject, Iterate. **Trigger** : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError, Synchronize, Parallelize. Pour plus d'informations concernant les liens, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **OUTPUT** : sortie standard d'un processus. Cette variable est une variable **After** et retourne une chaîne de caractères. **ERROROUTPUT** : sortie en erreur d'un processus. Cette variable est une variable **After** et retourne une chaîne de caractères. **EXIT_VALUE** : code de sortie d'un processus. Cette variable est une variable **After** et retourne un nombre entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.	

	Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Scénario : Echo ‘Hello World!'

Ce scénario est constitué d'un Job à deux composants permettant d'afficher un message dans la console de la vue **Run**.

Pour reproduire ce scénario, procédez comme suit :

1. Déposez un composant **tSystem** de la **Palette** dans l'espace de modélisation graphique.
2. Double-cliquez sur le **tSystem** pour ouvrir sa vue **Component**.

3. Sélectionnez l'option **Use Single Command** pour activer le champ **Command** et saisissez `"cmd /c echo Hello World!"`.
4. Dans la liste **Standard Output**, sélectionnez **to both console and global variable**.
5. Appuyez sur **F6** pour exécuter le Job.

```
Starting job tSystem_scenario at 17:26 08/10/2009.

[statistics] connecting to socket on port 3961
[statistics] connected
Hello World!
[statistics] disconnected
Job tSystem_scenario ended at 17:26 08/10/2009. [exit code=0]
```

Le Job exécute une commande `echo` et affiche la sortie dans la console de log de la vue **Run**.

Composants Talend MDM

Ce chapitre passe en revue les propriétés des principaux composants présents dans la famille **Talend MDM** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **Talend MDM** regroupe les composants qui lisent et écrivent des données maître dans le hub MDM.

tMDMBulkLoad

Propriétés du tMDMBulkLoad

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMBulkLoad** écrit en masse des données maître structurées XML dans le hub MDM.	
Objectif	Ce composant utilise le mode bulk pour écrire des données, de façon à ce que de gros lots de données ou des données très complexes puissent être chargé(e)s rapidement sur le serveur MDM.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	XML Field	Sélectionnez le nom de la colonne dans laquelle vous souhaitez écrire les données XML.
	URL	Saisissez l'URL d'accès au serveur MDM.
	Username et *Password*	Informations d'authentification de l'utilisateur du serveur MDM. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Version (déprécié)	Saisissez le nom de la Version des données maître à laquelle vous souhaitez vous connecter, en fonction de vos droits utilisateur. Laissez ce champ vide pour afficher la Version par défaut des données maître.
	Data Model	Saisissez le nom du modèle de données par rapport auquel vous souhaitez valider les données à écrire.
	Data Container	Saisissez le nom du conteneur de données contenant les données que vous souhaitez écrire.

	Entity	Saisissez le nom de l'entité contenant les données que vous souhaitez effacer.
	Type	Sélectionnez **Master** ou **Staging** pour spécifier sur quelle base de données effectuer l'action.
	Validate	Cochez cette case afin de valider les données que vous souhaitez écrire sur le serveur MDM, par rapport à des règles de validation définies pour le modèle de données courant. Pour le conteneur de données PROVISIONING, les vérifications de validation sont toujours effectuées sur les enregistrements entrants, que la case soit cochée ou non. Pour plus d'informations concernant la définition des règles de validation, consultez le *Guide utilisateur du Studio Talend*. *Si vous préférez un chargement plus rapide, ne cochez pas cette case.*
	Generate ID	Cochez cette case pour générer un identifiant aux données écrites. *Si vous préférez un chargement plus rapide, ne cochez pas cette case.*
	Commit size	Saisissez le nombre de lignes de chaque lot à écrire sur le serveur MDM.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Main. **Trigger** : Run If, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Main. **Trigger** : Run If, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError. Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant nécessite toujours un flux d'entrée afin de structurer des données en XML. Si vos données ne sont pas encore structurées XML, vous devez utiliser des composants tels que le **tWriteXMLField** afin de transformer ces données en XML. Pour plus d'informations concernant le **tWriteXMLField**, consultez *tWriteXMLField*.	

Améliorer les chargements de données MDM de masse

Les informations ci-dessous concernent uniquement MDM utilisé avec eXist (déprécié).

L'analyse XML est un processus qui consomme beaucoup de ressources du processeur et de mémoire et qui n'est pas vraiment compatible avec des ensembles de données particulièrement volumineux. Dans *Scénario : Charger des enregistrements dans une entité métier*, le scénario donné en exemple pour le composant **tMDMBulkLoad**

possède des limitations parce qu'il ne fonctionne pas avec un ensemble de données très volumineux, pour le moment au moins.

Un scénario alternatif dans lequel vous pouvez traiter l'ensemble de données par itérations de chargement de masse peut être créé comme suit :

Dans un tel scénario, le composant **tMDMBulkLoad** attend des données XML en entrée. Vous devez formater manuellement ces données entrantes afin de les faire correspondre au schéma de l'entité défini dans la perspective **MDM** du *Studio Talend*. La plupart du temps, les données que vous souhaitez importer sont au format plat et vous devez les transformer en XML.

Puisque l'analyse XML consomme beaucoup de mémoire, vous pouvez régler ce problème en séparant votre fichier source en plusieurs fichiers utilisant le composant **tAdvancedFileOutputXML**. Pour ce faire, sélectionnez l'option **Split output in several files** dans la vue **Advanced settings** du composant puis configurez les lignes dans chaque fichier de sortie via une variable de contexte (*context.chunkSize*), par exemple.

Le schéma XML que vous devez définir dans l'éditeur XML du composant doit être une correspondance exacte de l'entité métier définie dans la perspective **MDM** du *Studio Talend*. Le schéma XML dans l'éditeur doit représenter

un seul élément racine `<root>` qui contient tous les autres éléments, afin que vous puissiez effectuer une boucle sur chaque élément. Le chemin d'accès au fichier doit être défini dans un dossier temporaire.

Utilisez un composant **tFileList** pour lire tous les fichiers XML ayant été créés. Ce composant vous permet de paralléliser le processus. Connectez-le à un composant **tFileInputXML** à l'aide d'un lien **Iterate**.

Pour le lien **Iterate**, il est recommandé de configurer autant de threads qu'il y a de cœurs physiques à votre ordinateur. Vous pouvez faire ceci en utilisant `Runtime.getRuntime().availableProcessors()`

Le composant **tFileInputXML** va lire les données des fichiers XML que vous avez créés, en définissant une boucle sur les éléments et en récupérant tous les nœuds déjà formatés en XML. Vous devez ensuite cocher la case **Get Nodes**.

Enfin, vous devez configurer le composant **tMDMBulkLoad** comme suit :

Soyez sûr d'avoir configuré la taille de commit avec la même valeur que celle définie dans le composant **tAdvancedfileOutputXML**, la variable de contexte *context.chunkSize*.

Le composant **tFileDelete** dans ce scénario va supprimer toutes les données temporaires à la fin du Job.

Scénario : Charger des enregistrements dans une entité métier

Ce scénario décrit un Job qui charge des enregistrements dans l'entité métier *ProductFamily* définie par un modèle de données spécifique dans le hub MDM.

Prérequis pour ce Job :

- Le conteneur de données *Product* : ce conteneur de données est utilisé pour séparer le domaine de données maître des produits des autres domaines de données maître.
- Le modèle de données *Product* : ce modèle de données est utilisé pour définir les attributs, les règles de validation, les droits d'accès des utilisateurs et les relations des entités en rapport. Il définit les attributs de l'entité métier *ProductFamily*.
- L'entité métier *ProductFamily* : cette entité métier contient les attributs *Id* et *Name*, définis dans le modèle de données *Product*.

Pour plus d'informations concernant la création d'un conteneur de données, d'un modèle de données et d'une entité métier ainsi que de ses attributs, consultez le *Guide utilisateur du Studio Talend*.

Le Job dans ce scénario utilise trois composants.

- Un **tFixedFlowInput** : ce composant génère les enregistrements à charger dans l'entité métier *ProductFamily*. Généralement, les enregistrements à charger sont volumineux et stockés dans un fichier spécifique, cependant, pour faciliter la reproduction de ce scénario, ce Job utilise un **tFixedFlowInput** afin de générer quatre exemples d'enregistrements.
- Un **tWriteXMLField** : ce composant transforme les données d'entrée pour leur donner une structure XML.
- Un **tMDMBulkLoad** : ce composant écrit en masse les données d'entrée dans l'entité métier *ProductFamily* et génère une valeur d'ID pour chaque donnée de l'enregistrement.

Pour reproduire ce scénario, procédez comme suit :

- Déposez un **tFixedFlowInput**, un **tWriteXMLField** et un **tMDMBulkLoad** de la **Palette** dans l'espace de modélisation graphique.
- Cliquez-droit sur le **tFixedFlowInput** afin d'ouvrir son menu contextuel.
- Sélectionnez **Row > Main** pour relier le **tFixedFlowInput** au composant suivant à l'aide d'un lien **Main**.
- Répétez l'opération pour lier les autres composants.
- Double-cliquez sur le **tFixedFlowInput** afin d'ouvrir sa vue **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

- Dans l'éditeur du schéma, cliquez sur le bouton **[+]** pour ajouter une ligne.
- Cliquez sur la nouvelle et saisissez son nom : *family*.
- Cliquez sur **OK**.
- Dans la zone **Mode** de la vue **Basic settings**, sélectionnez l'option **Use inline table**.
- Sous le tableau, cliquez sur le bouton **[+]** quatre fois afin d'ajouter quatre lignes au tableau.
- Dans le tableau, cliquez sur chacune des lignes ajoutées et saisissez leur nom, entre guillemets : *Shirts*, *Hats*, *Pets*, *Mugs* (Chemises, chapeaux, vêtements pour animaux, tasses).
- Double-cliquez sur le composant **tWriteXMLField** afin d'ouvrir sa vue **Basic settings**.

- Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir l'éditeur du schéma, dans lequel vous pouvez ajouter une ligne en cliquant sur le bouton **[+]**.

- Cliquez sur la nouvelle ligne, dans la partie droite de l'éditeur du schéma et saisissez le nom de la colonne de sortie dans laquelle vous souhaitez écrire le contenu XML. Dans cet exemple, saisissez *xmlRecord*.
- Cliquez sur **OK** pour valider le schéma de sortie et fermer l'éditeur du schéma.
- Dans la boîte de dialogue qui s'ouvre, cliquez sur **OK** afin de propager le schéma au composant suivant.
- Dans la vue **Basic settings**, cliquez sur le bouton **[...]** à côté du champ **Configure Xml Tree** pour ouvrir l'interface permettant de créer la structure XML.

- Dans la zone **Link Target**, cliquez sur *rootTag* et renommez le champ *ProductFamily*, qui est le nom de l'entité métier utilisée dans ce scénario.

- De la zone **Linker source**, glissez-déposez *family* dans *ProductFamily*, dans la zone **Link target**.

  Une boîte de dialogue s'affiche et vous demande quel type d'opération vous souhaitez effectuer.

- Sélectionnez **Create as sub-element of target node** afin de créer un sous-élément du nœud *ProductFamily*. L'élément *family* apparaît sous le noeud *ProductFamily*.

- Dans la zone **Link target**, cliquez sur le nœud *family* et renommez-le *Name*, qui est l'un des attributs de l'entité métier *ProductFamily*.

- Cliquez-droit sur le nœud *Name* et sélectionnez **Set As Loop Element** dans le menu contextuel.

- Cliquez sur **OK** afin de valider la structure XML que vous avez définie.

- Double-cliquez sur le **tMDMBulkLoad** afin d'ouvrir sa vue **Basic settings**.

- Dans la liste **XML Field**, sélectionnez *xmlRecord*.

- Dans le champ **URL**, saisissez l'URL dédiée au chargement de masse, entre guillemets : par exemple, *http://localhost:8080/datamanager/loadServlet*.

- Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant et votre mot de passe de connexion au serveur MDM.

- Dans les champs **Data Model** et **Data Container**, saisissez les noms correspondants au modèle de données et au conteneur de données que vous souhaitez utiliser, *Product* pour les deux champs, dans ce scénario.

- Dans le champ **Entity**, saisissez le nom de l'entité métier dont les enregistrements doivent être chargés. Dans cet exemple, saisissez *ProductFamily*.

- Cochez la case **Generate ID** afin de générer des valeurs d'ID pour les enregistrements à charger.

- Appuyez sur **F6** pour exécuter le Job.

- Connectez-vous à *Talend MDM Web User Interface* pour vérifier les nouveaux enregistrements de l'entité métier *ProductFamily*.

Browse Records
Search panel
Entity : Product Family
Search criteria :
Id (sequence)
contains the word(s)
*
AND
OR
Search
New Record
Page 1 of 1
Number of lines per page : 20
Displaying records: 1 - 4 of 4
Id (sequence)
Name
13
Mugs
14
Pets
15
Hats
16
Shirts
Delete
Send to Trash

tMDMClose

Propriétés du tMDMClose

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMClose** ferme une connexion ouverte au serveur MDM.	
Objectif	Ce composant est utilisé pour fermer un connexion ouverte au serveur MDM après l'exécution du sous-job précédent.	
Basic settings	*Component List*	S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant **tMDMConnection** dans la liste.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Dynamic settings	Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion au serveur MDM parmi celles prévues dans votre Job. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** devient inaccessible dans la vue **Basic settings**. Pour plus d'informations concernant les **Dynamic settings** et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.	
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être utilisé avec le **tMDMConnection**. Notez que les composants **tMDMCommit** et **tMDMRollback** comprennent également une option, dans leur onglet **Basic settings**, permettant de fermer une connexion.	

Scénario associé

Pour un scénario associé, consultez *tMDMDelete*.

tMDMCommit

Propriétés du tMDMCommit

Famille de composant	Talend MDM	
Fonction	Le **tMDMCommit** commite définitivement toutes les modifications apportées à une base de données dans le cadre d'une transaction dans MDM.	
Objectif	Ce composant est utilisé pour contrôler le moment où, dans un Job MDM, les modifications apportées à la base de données sont commitées, dans le cadre d'une transaction MDM, par exemple pour éviter des commits partiels lorsqu'une erreur survient. Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction **Auto Commit** disponible dans le composant de connexion.	
Basic settings	*Component List*	Sélectionnez le composant **tMDMConnection** pour lequel effectuer l'action de commit.
	Close Connection	Cochez cette case pour fermer la session de la connexion au serveur MDM après avoir commité les modifications. Notez que, même si la case n'est pas cochée, la connexion ne peut toujours pas être utilisée par le sous-job suivant, sauf si le mode Auto-Commit est activé.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau des composants.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec les composants **tMDM Connection**, **tMDMRollback**. **tMDMSP**, **tMDMViewSearch**, **tMDMInput**, **tMDMDelete**, **tMDMRouteRecord**, **tMDMOutput** et **tMDMClose**.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Supprimer des données maître d'un hub MDM*.

tMDMConnection

Propriétés du tMDMConnection

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMConnection** ouvre une connexion au serveur MDM afin d'en permettre la réutilisation dans la transaction en cours.	
Objectif	Ce composant est utilisé pour ouvrir une connexion à un serveur MDM afin de pouvoir la réutiliser dans le(s) sous-job(s) suivants et éviter de spécifier les informations de connexion dans chaque composant.	
Basic settings	*URL*	Saisissez l'URL pour accéder au serveur MDM.
	Username et *Password*	Informations d'authentification de l'utilisateur du serveur MDM. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Version (déprécié)	Saisissez le nom de la Version des données maître auxquelles vous souhaitez vous connecter Laissez ce champ vide si vous souhaitez afficher la Version par défaut des données maître.
	Auto Commit	Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit. Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être utilisé avec les composants **tMDMCommit**, **tMDMRollback** **tMDMSP**, **tMDMViewSearch**, **tMDMInput**, **tMDMDelete**, **tMDMRouteRecord**, **tMDMOutput** et **tMDMClose**.	

Scénario associé

Pour un scénario associé, consultez *tMDMDelete*.

tMDMDelete

Propriétés du tMDMDelete

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMDelete** supprime des enregistrements de données d'entités spécifiques du hub MDM.	
Objectif	Ce composant efface des données maître du hub MDM.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant **tMDMConnection** adéquat à partir de la liste **Component** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	URL	Saisissez l'URL d'accès au serveur MDM.
	Username et *Password*	Informations d'authentification de l'utilisateur du serveur MDM. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Version (déprécié)	Saisissez le nom de la Version des données maître à laquelle vous souhaitez vous connecter, en fonction de vos droits utilisateur. Laissez ce champ vide pour afficher la Version par défaut des données maître.
	Entity	Saisissez le nom de l'entité contenant les données que vous souhaitez effacer.
	Data Container	Saisissez le nom du conteneur de données contenant les données que vous souhaitez effacer.

	Type	Sélectionnez **Master** ou **Staging** pour spécifier sur quelle base de données effectuer l'action.
	Use multiple conditions	Cochez cette case afin de filtrer les données maître à supprimer, en respectant certaines conditions. **Xpath** : Saisissez entre guillemets le chemin d'accès et le nœud XML auquel vous souhaitez appliquer la condition. **Function** : Sélectionnez dans la liste la condition à utiliser. **Value** : Saisissez entre guillemets la valeur à utiliser. **Predicate** : Sélectionnez un prédicat si vous utilisez plus d'une condition.
	Keys (in sequence order)	Spécifiez le(s) champ(s) (dans l'ordre de la séquence) composant la clé lorsque l'entité possède une clé multiple.
	Fire Delete event	Cette case est disponible uniquement si vous sélectionnez l'option **Master** dans la liste **Type**. Cochez cette case pour qu'un événement Delete soit généré et écrit dans le rapport **UpdateReport**, qui fournit un rapport complet sur toutes les actions de création, de mise à jour ou de suppression dans les données maître. • **Source Name** : saisissez, entre guillemets, le nom de la source générant l'événement de suppression. • **Invoke before deleting process** : cochez cette case pour permettre l'exécution d'un Processus **Before Deleting** pour vérifier l'événement Delete. Pour plus d'informations concernant les rapports **UpdateReport** et les types de Processus, consultez le *Guide utilisateur du Studio Talend*.
	Logical delete	Cochez cette case afin d'envoyer les données maître dans la Corbeille, et renseignez le chemin d'accès à la Corbeille (**Recycle bin path**). Une fois dans la Corbeille, les données maître peuvent être supprimées définitivement ou restaurées. Si vous laissez la case décochée, les données maître seront supprimées définitivement.
	Die on error	Cochez cette case pour passer la ligne en erreur et terminer le processus afin d'obtenir des lignes ne contenant aucune erreur. Si nécessaire, vous pouvez récupérer les lignes en erreur en utilisant un lien de type **Row > Rejects**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Si la case **Use multiple conditions** est cochée, le **tMDMDelete** peut fonctionner en standalone. Sinon, il requiert un lien d'entrée.	

Scénario : Supprimer des données maître d'un hub MDM

Ce scénario décrit un Job à six composants qui supprime un enregistrement de données spécifié du référentiel XML MDM. Il utilise les composants **tMDMCommit** et **tMDMRollback** afin de ne propager les modifications à la base de données uniquement lorsque le Job est correctement exécuté.

Déposer et relier les composants

1. Déposez un composant **tMDMConnection**, un **tMDMCommit**, un **tMDMClose**, un **tMDMInput**, un **tMDMDelete** et un **tMDMRollback** de la famille **Talend MDM** de la **Palette** dans l'espace de modélisation graphique.

2. Reliez le **tMDMInput** au **tMDMDelete** à l'aide d'un lien **Row** > **Main**.

3. Reliez le composant **tMDMConnection** au **tMDMInput** à l'aide d'un lien **Trigger** > **OnSubjobOk**.

4. Reliez le **tMDMDelete** au **tMDMCommit**, le **tMDMCommit** au **tMDMClose** et le **tMDMRollback** au **tMDMClose** à l'aide de liens **Trigger** > **OnComponentOk**.

5. Reliez le **tMDMDelete** au **tMDMRollback** à l'aide d'un lien **Trigger** > **OnComponentError**.

Configurer la connexion au serveur MDM

Dans ce scénario, un composant **tMDMConnection** est utilisé pour ouvrir une connexion au serveur MDM et la réutiliser ultérieurement dans le sous-job suivant, qui supprime l'enregistrement de données spécifié.

1. Double-cliquez sur le **tMDMConnection** afin d'afficher sa vue **Basic settings** et définir ses propriétés de base.

2. Dans le champ **URL**, saisissez entre guillemets l'URL du serveur MDM, par exemple : *"http://localhost:8180/talend/TalendPort"*.

3. Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant et votre mot de passe pour vous connecter au serveur MDM.

4. Dans le champ **Version** (déprécié), saisissez entre guillemets le nom de la Version des données maître à laquelle vous souhaitez accéder. Laissez ce champ vide pour accéder à la Version par défaut des données maître.

5. Double-cliquez sur le **tMDMCommit** pour afficher sa vue **Basic settings**.

Ce composant commite les modifications apportées à la base de données lorsque le sous-job est exécuté avec succès.

6. Dans la liste **Component List**, sélectionnez le composant de connexion au serveur que vous souhaitez fermer, si vous avez configuré plusieurs connexions à des serveurs MDM. Dans ce scénario, une seule connexion au serveur est ouverte, utilisez donc simplement le paramètre par défaut.

7. Décochez la case **Close Connection** si elle est cochée. Dans cet exemple, le composant **tMDMClose** ferme la connexion au serveur MDM.

8. Double-cliquez sur le composant **tMDMRollback** pour afficher sa vue **Basic settings**.

Ce composant effectue un rollback des modifications et rétablit l'état précédent de la base de données si le sous-job échoue.

9. Dans la liste **Component List**, sélectionnez le composant de connexion au serveur que vous souhaitez fermer, si vous avez configuré plusieurs connexions à des serveurs MDM. Dans ce scénario, une seule connexion au serveur est ouverte, utilisez donc simplement le paramètre par défaut.

10. Décochez la case **Close Connection** si elle est cochée. Dans cet exemple, le composant **tMDMClose** ferme la connexion au serveur MDM.

11. Double-cliquez sur le composant **tMDMClose** pour afficher sa vue **Basic settings** et définir ses propriétés de base.

Le composant **tMDMClose** est utilisé pour fermer la connexion après l'exécution réussie du Job. Vous pouvez également fermer la connexion en cochant la case **Close Connection** dans les composants **tMDMCommit** et **tMDMRollback**. Dans un souci de démonstration, un **tMDMClose** est utilisé dans ce scénario.

12. Dans la liste **Component List**, sélectionnez le composant dont vous souhaitez fermer la connexion au serveur, si vous avez configuré plus d'une connexion au serveur MDM. Dans ce scénario, une seule connexion est ouverte, utilisez simplement la configuration par défaut.

Configurer la récupération de données

1. Double-cliquez sur le **tMDMInput** pour afficher sa vue **Basic settings** et définir ses propriétés de base.

2. Dans la liste **Property Type**, sélectionnez **Built-in** afin de renseigner les champs manuellement.

   Si vous avez stocké vos informations de connexion au serveur MDM dans le référentiel des métadonnées, sélectionnez **Repository** dans la liste et les champs seront automatiquement renseignés.

3. Dans la liste **Schema**, sélectionnez **Built-in** et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir une boîte de dialogue.

   Vous pouvez définir la structure des données maître que vous souhaitez lire dans le hub MDM.

4. Les données maître sont collectée dans trois colonnes de type **String** : *Id*, *Name* et *Price*. Cliquez sur **OK** pour fermer la boîte de dialogue et passer à l'étape suivante.

5. Cochez la case **Use an existing connection** et, dans la liste **Component List** qui apparaît, sélectionnez le composant que vous avez configuré pour ouvrir votre connexion au serveur MDM.

   Dans ce scénario, une seule connexion au serveur MDM existe, utilisez donc la sélection par défaut.

6. Dans le champ **Entity**, saisissez entre guillemets le nom de l'entité métier contenant les enregistrements de données que vous souhaitez lire. Ici, saisissez le nom de l'entité *Product*.

7. Dans le champ **Data Container**, saisissez entre guillemets le nom du conteneur de données contenant les données maître que vous souhaitez lire. Dans cet exemple, utilisez le conteneur *Product*.

   La case **Use multiple conditions** est cochée par défaut.

8. Dans la table **Operations**, définissez comme suit les conditions pour filtrer les données maître que vous souhaitez supprimer :

   - Cliquez sur le bouton **[+]** pour ajouter une nouvelle ligne.
   - Dans la colonne **Xpath**, saisissez entre guillemets le chemin Xpath et la balise du nœud XML sur lequel appliquer le filtre. Dans cet exemple, vous travaillez avec l'entité **Agency**, saisissez donc *"Product/Name"*.
   - Dans la colonne **Function**, sélectionnez la fonction que vous souhaitez utiliser. Dans ce scénario, utilisez la fonction *Contains*.
   - Dans la colonne **Value**, saisissez la valeur de votre filtre. Filtrez les données maître dont le nom contient *mug*.

9. Dans la vue **Component**, cliquez sur **Advanced settings** afin de configurer les paramètres avancés.

10. Dans le champ **Loop XPath query**, saisissez entre guillemets la structure et le nom du nœud XML sur lequel la boucle doit être effectuée.

11. Dans la table **Mapping** et dans la colonne **XPath query**, saisissez entre guillemets le nom de la balise XML dans laquelle vous souhaitez collecter les données maître, à côté du nom de la colonne de sortie correspondante.

Configurer la suppression de l'enregistrement de données

1. Dans l'espace de modélisation graphique, double-cliquez sur le composant **tMDMDelete** afin d'afficher sa vue **Basic settings** et définir ses propriétés de base.

2. Dans la liste **Schema**, sélectionnez **Built-in** puis cliquez sur le bouton **[...]** à côté du champ **Edit Schema** afin de décrire la structure des données maître dans le hub MDM.

3. Cliquez sur le bouton **[+]** de droite afin d'ajouter une colonne de type **String**. Nommez cette colonne *outputXML*. Cliquez sur **OK** pour fermer la boîte de dialogue et passer à l'étape suivante.

4. Cochez la case **Use an existing connection**, puis, dans la liste **Component List** qui apparaît, sélectionnez le composant que vous avez configuré pour ouvrir votre connexion au serveur MDM.

   Dans ce scénario, une seule connexion au serveur MDM existe, utilisez la sélection par défaut.

5. Dans le champ **Entity**, saisissez le nom de l'entité métier contenant les données maître que vous souhaitez supprimer, l'entité *Product* dans cet exemple.

6. Dans le champ **Data Container**, saisissez le nom du conteneur de données contenant les données à supprimer, *Product* dans cet exemple.

7. Dans le tableau **Keys**, cliquez sur le bouton **[+]** pour ajouter une nouvelle ligne. Dans la colonne **Keys**, sélectionnez la colonne contenant la clé de l'entité *Product*. Dans ce scénario, la clé de l'entité *Product* est définie dans le champ *Id*.

   Si l'entité possède plusieurs clés, ajoutez autant de ligne que nécessaire pour les clés et sélectionnez-les en ordre séquentiel.

8. Si nécessaire, cochez la case **Fire Delete event**, pour que l'événement de suppression soit généré et écrit dans le rapport **UpdateReport**.

9. Cochez la case **Logical delete** si vous ne souhaitez pas supprimer les données maître de manière permanente. Cela envoie les données supprimées dans la corbeille. Une fois dans la corbeille, les données maître peuvent être restaurées ou supprimées définitivement. Si vous laissez la case décochée, les données maître seront supprimées définitivement.

10. Renseignez le champ **Recycle bin path**. Laissez le chemin d'accès par défaut, sauf si le chemin d'accès à votre corbeille est différent de celui par défaut. Dans ce cas, saisissez le chemin d'accès à votre corbeille.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job et vous assurer que tous les paramètres configurés seront pris en compte.

2. Appuyez sur **F6** pour exécuter votre Job.

Les données maître dont le nom contient "mug" ont été supprimées et envoyées dans la corbeille.

tMDMInput

Propriétés du tMDMInput

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMInput** lit les données maître du hub MDM.	
Objectif	Ce composant lit les données maître dans un hub MDM et rend possible le traitement de ces données.	
Basic settings	*Property Type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Use an existing connection	Cochez cette case et sélectionnez le composant **tMDMConnection** adéquat à partir de la liste **Component** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	URL	Saisissez l'URL d'accès au serveur MDM.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur du serveur MDM. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Version (déprécié)	Saisissez le nom de la Version des données maître à laquelle vous souhaitez vous connecter, en fonction de vos droits utilisateur. Laissez ce champ vide pour afficher la Version par défaut des données maître.

	Entity	Saisissez le nom de l'entité contenant les données que vous souhaitez effacer.
	Data Container	Saisissez le nom du conteneur de données contenant les données que vous souhaitez effacer.
	Type	Sélectionnez **Master** ou **Staging** pour spécifier sur quelle base de données effectuer l'action.
	Use multiple conditions	Cochez cette case pour filtrer les données maître à l'aide de conditions. **Xpath** : saisissez, entre guillemets, le nœud XML sur lequel appliquer la condition. **Function** : sélectionnez la condition à utiliser. Selon le type de champ vers lequel pointe le XPath, seuls certains opérateurs s'appliquent. Par exemple, si le champ est un booléen, seuls les opérateurs *Equal* et *Not Equal* s'appliquent. Les opérateurs suivants sont disponibles : • *Contains* : Retourne un résultat contenant le ou les mot(s) saisi(s). • *Joins With* : Retourne un résultat fonctionnant comme un opérateur de jointure. • *Starts With* : Retourne un résultat commençant par la chaîne de caractères saisie. • *Strict Contains* : Retourne un résultat contenant l'expression régulière exacte saisie. S'applique uniquement aux bases de données XML. • *Equal* : Retourne un résultat correspondant au booléen saisi, *True* ou *False*. • *Not Equal* : Retourne un résultat dont la valeur est différente du booléen saisi, *True* ou *False*. • *is greater than* : Retourne un résultat supérieur à la valeur numérique saisie. S'applique uniquement aux champs contenant des nombres. • *is greater or equal* : Retourne un résultat supérieur ou égal à la valeur numérique saisie. S'applique uniquement aux champs contenant des nombres. • *is lower than* : Retourne un résultat inférieur à la valeur numérique saisie. S'applique uniquement aux champs contenant des nombres. • *is lower or equal* : Retourne un résultat inférieur ou égal à la valeur numérique saisie. S'applique uniquement aux champs contenant des nombres. • *whole content contains* : Effectue une recherche plein texte dans tous les champs de l'entité. Pour les bases de données SQL, une recherche "Starts with" est effectuée. Pour les bases de données XML, une recherche "Contains" est effectuée. • *contains a word like* : Effectue une recherche floue afin de retourner un mot similaire au mot saisi. • *is empty or null* : Retourne un résultat dans lequel le champ est vide ou retourne une valeur nulle. **Value** : saisissez, entre guillemets, la valeur souhaitée. Notez que si la valeur contient des caractères XML spéciaux, comme /, vous devez saisir la valeur entre guillemets simples également ('"ABC/XYZ"'), sinon la valeur est considérée comme XPath. **Predicate** : sélectionnez un prédicat lorsque vous utilisez plusieurs conditions. Les prédicats suivants sont disponibles :

		• *Default* : Interprété comme un *and*. • *or* : L'une des conditions s'applique. • *and* : Toutes les conditions s'appliquent. Les autres prédicats sont réservés pour usage ultérieur et peuvent être sujets à des comportements inattendus. Si vous décochez cette case, vous avez la possibilité de sélectionner des ID particuliers devant être affichés dans la colonne **ID value** du tableau **IDS**. Si vous décochez la case **Use multiple conditions**, l'option **Batch Size** de l'onglet **Advanced settings** ne sera plus disponible.
	Skip Rows	Renseignez le nombre de lignes à ignorer.
	Max Rows	Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n'est lue ou traitée.
	Die on error	Cochez cette case pour passer la ligne en erreur et terminer le processus afin d'obtenir des lignes ne contenant aucune erreur. Si nécessaire, vous pouvez récupérer les lignes en erreur en utilisant un lien de type **Row > Rejects**.
Advanced settings	*Batch Size*	Nombre de lignes dans chaque lot traité. Cette option n'apparaît pas si vous avez décoché la case **Use multiple conditions** dans l'onglet **Basic settings**.
	Loop XPath query	Nœud de l'arborescence XML sur lequel est basée la boucle.
	Mapping	**Column** : reflète le schéma tel qu'il est défini dans l'éditeur **Edit Schema**. **XPath query** : Saisissez les champs à extraire de la structure XML d'entrée. **Get Nodes** : Cochez cette case pour récupérer les balises XML entourant les données avec celles-ci.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est utilisé comme composant de début. Il nécessite un flux de sortie.	

Scénario : Lire les données maître d'un hub MDM

Ce scénario décrit un Job à deux composants qui lit les données maître dans un serveur MDM et les affiche dans la console de log.

- A partir de la **Palette**, glissez les composants **tMDMInput** de la famille **Talend MDM** et **tLogRow** de la famille **Logs & Errors** dans l'espace de modélisation.
- Reliez-les via un lien de type **Row Main**.
- Double-cliquez sur le composant **tMDMInput** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.

- Dans la liste **Property Type**, sélectionnez **Built-In** et renseignez les champs manuellement. Si vous avez stocké les informations de connexion au MDM dans les métadonnées du Repository, sélectionnez **Repository** dans la liste et les champs suivants seront renseignés automatiquement.
- Dans la liste **Schema**, sélectionnez **Built-In** et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour décrire la structure des données maître que vous souhaitez lire sur le serveur MDM.

- Les données maître sont récupérées dans un schéma à trois colonnes de type **String** : *ISO2Code*, *Name*, *Currency*. Cliquez sur **OK**.
- Dans le champ **URL**, saisissez entre guillemets l'URL de connexion au serveur MDM.
- Dans les champs **Username** et **Password**, saisissez l'identifiant et le mot de passe de connexion au serveur MDM.
- Dans le champ **Version** (déprécié), saisissez entre guillemets le nom de la Version des données maître à laquelle vous avez souhaitez accéder. Laissez le champ vide pour afficher la Version par défaut.
- Dans le champ **Entity**, saisissez entre guillemets le nom de l'entité métier contenant les données maître que vous souhaitez lire.
- Dans le champ **Data Container**, saisissez entre guillemets le nom du conteneur de données contenant les données maître que vous souhaitez lire.
- Dans la vue **Component**, cliquez sur l'onglet **Advanced settings** pour paramétrer les propriétés avancées du composant.

- Dans le champ **Loop XPath query**, saisissez entre guillemets l'arborescence et le nom du nœud XML sur lequel effectuer la boucle.
- Dans le tableau **Mapping**, les colonnes du schéma du composant apparaissent sous la colonne **Column**. Saisissez entre guillemets le nom de la balise XML dans laquelle récupérer les données maître dans le champ **XPath query** correspondant à la colonne du schéma de sortie.
- Dans l'espace de modélisation, double-cliquez sur le composant **tLogRow** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.
- Cliquez sur **Edit schema** et assurez-vous que le schéma est récupéré du composant précédent. Sinon, cliquez sur **Sync Columns** pour récupérer le schéma du composant.
- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job docPays at 16:18 03/11/2009.
[statistics] connecting to socket on port 3916
[statistics] connected
AD|Andorra|[EUR]
AE|United Arab Emirates|[AED]
AF|Afghanistan|[AFN]
AG|Antigua and Barbuda|[XCD]
AI|Anguilla|[XCD]
AL|Albania|[ALL]
AM|Armenia|[AMD]
AN|Netherlands Antilles|[ANG]
AO|Angola|[AOA]
AR|Argentina|[ARS]
AS|American Samoa|[USD]
AT|Austria|[EUR]
AU|Australia|[AUD]
AW|Aruba|[AWG]
```

La liste des différents pays, ainsi que leur code et devise est retournée dans la console de la vue **Run**.

tMDMOutput

Propriétés du tMDMOutput

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMOutput** écrit des données maître dans un hub MDM.	
Objectif	Ce composant écrit des données maître sur le serveur MDM.	
Basic settings	*Property Type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Input Schema et *Edit schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	Build the document	Cochez cette case si vous souhaitez construire le document à partir d'un schéma plat. Dans ce cas, double-cliquez sur le composant et mappez votre schéma dans la boîte de dialogue qui s'ouvre. Si la case est décochée, vous devez sélectionner, dans la liste **Predefined XML document**, une colonne de votre schéma contenant le document.
	Result of the XML serialization	Sélectionnez le nom de la colonne de sortie dans laquelle vous souhaitez écrire les données XML.
	Use an existing connection	Cochez cette case pour utiliser un composant **tMDMConnection** déjà configuré.
	URL	Saisissez l'URL d'accès au serveur MDM.
	Username et *Password*	Saisissez les informations d'authentification de l'utilisateur du serveur MDM.

		Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres. Cet utilisateur doit avoir le bon rôle dans la solution MDM de Talend, c'est-à-dire qu'il puisse se connecter via un Job ou tout autre appel de service Web. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*.
	Version (déprécié)	Saisissez le nom de la Version des données maître à laquelle vous souhaitez vous connecter, en fonction de vos droits utilisateur. Laissez ce champ vide pour afficher la Version par défaut des données maître.
	Data Model	Saisissez le nom du modèle de données en comparaison duquel les données à écrire seront validées.
	Data Container	Saisissez le nom du conteneur de données dans lequel vous souhaitez écrire les données maître. Ce conteneur de données doit avoir été créé.
	Type	Sélectionnez **Master** ou **Staging** pour spécifier sur quelle base de données effectuer l'action.
	Return Keys	**Columns corresponding to IDs in order** : dans l'ordre séquentiel, définissez les colonnes de sortie qui stockeront les valeurs des clés retournées (clés primaires) des éléments à créer.
	Is Update	Cochez cette case pour mettre à jour les champs modifiés. Si vous ne cochez pas cette case, la totalité des champs sera remplacée par les champs modifiés.
	Fire Create/Update event	Cochez cette case pour ajouter les actions effectuées à un rapport de modifications. **Source Name** : Saisissez, entre guillemets, le nom de l'application à partir de laquelle les modifications sont apportées. **Enable verification by "before saving" transformer** : Cochez cette case pour vérifier l'enregistrement qui vient d'être ajouté avant qu'il ne soit enregistré.

	Use partial update	Cochez cette case si vous devez mettre à jour des éléments multi-occurences (attributs) d'une entité existante à partir du contenu d'un flux XML d'entrée. Une fois la case cochée, vous devez configurer les paramètres ci-dessous :
		
		- **Pivot** : saisissez le chemin XPath vers les sous-éléments multi-occurences dans lesquels des données doivent être ajoutées ou remplacées dans l'entité de votre choix. Par exemple, vous souhaitez ajouter un sous-élément fils à une entité existante, comme suit :
		
		`<Person>`
		`  <Id>1</Id>`
		`  <Addresses>`
		`  <Address>`
		`  <Type>office</Type>`
		`  (...address elements are here....)`
		`  </Address>`
		`  <Address>`
		`  <Type>home</Type>`
		`  (...address elements are here....)`
		`  </Address>`
		`  <Addresses>`
		`</Person>`
		
		le chemin XPath saisi dans le champ **Pivot** doit être comme suit :
		
		`Person/Children/Child`
		
		et la case **Overwrite** devrait être décochée.

		Et, si vous devez remplacer un sous-élément fils dans une entité existante :
		
		`<Person>`
		`  <Id>1</Id>`
		`  <Addresses>`
		`  <Address>`
		`  <Type>office</Type>`
		`  (...address elements are here....)`
		`  </Address>`
		`  <Address>`
		`  <Type>home</Type>`
		`  (...address elements are here....)`
		`  </Address>`
		`  <Addresses>`
		`</Person>`
		
		le chemin XPath saisi dans le champ **Pivot** doit être comme suit : `/Person/Addresses/Adress` la case **Overwrite** doit être cochée et la champ **Key** configuré comme suit : `/Type`.
		
		Dans cet exemple, nous supposons que l'entité MDM a seulement une adresse de type *office*, ainsi cette adresse de type *office* sera remplacée et l'adresse de type *home* sera ajoutée.
		
		- **Overwrite** : cochez cette case si vous devez remplacer ou mettre à jour les sous-éléments originaux par les sous-éléments d'entrée. Laissez cette case décochée si vous souhaitez ajouter un sous-élément.
		
		- **Key** : saisissez le chemin XPath relatif au pivot qui permettra de mettre en correspondance le sous-élément du flux XML d'entrée avec le sous-élément de l'entité. Si une clé n'est pas fournie, tous les sous-éléments de l'entité ayant un XPath qui correspond à celui du sous-élément du flux XML d'entrée seront remplacés. Si plus d'un sous-élément correspond à la clé, MDM met à jour le premier qu'il rencontre. Si aucun sous-élément ne correspond à la clé, elle est ajoutée à la fin de la collection.
		
		-**Position** : saisissez un chiffre pour indiquer la position après laquelle les nouveaux éléments (ceux qui ne correspondent pas à la clé) seront ajouté. Si vous ne renseignez pas ce champ, les nouveau éléments seront ajoutés à la fin.

	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Rejects**.
Advanced settings	*Extended Output*	Cochez cette case pour commiter les données maître par lot. **Rows to commit** : saisissez le nombre de lignes par lot.
	Configure Xml Tree	Ouvrez l'interface d'aide à la création de la structure XML des données maître à écrire.
	Group by	Sélectionnez la colonne à utiliser pour regrouper les données maître.
	Create empty element if needed	Cette case est cochée par défaut. Si le contenu de la colonne **Related Column** de l'interface permettant de créer la structure XML est nul, ou si aucune colonne n'est associée au nœud XML, cette option créera une balise ouvrante et une balise fermante aux endroits prévus.
	Advanced separator (for number)	Cochez cette case pour modifier les séparateurs utilisés par défaut dans les nombres : - **Thousands separator** : saisissez entre guillemets le séparateur à utiliser pour les milliers. - **Decimal separator** : saisissez entre guillemets le séparateur à utiliser pour les décimales.
	Generation mode	Sélectionnez le mode de génération correspondant à votre mémoire disponible. Les modes disponibles sont : • Lent et consommateur de mémoire (**Slow and memory-consuming - Dom4J**).

		Cette option vous permet d'utiliser Dom4J pour traiter des fichiers XML très complexes. • Rapide et peu consommateur de mémoire (**Fast with low memory consumption (SAX)**).
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour écrire un fichier et séparer les champs à l'aide d'un séparateur spécifique.	

Scénario : Ecrire des données maître dans un hub MDM

Ce scénario décrit un Job à deux composants qui génère un enregistrement de données, transforme ces données en XML et les charge dans l'entité métier définie sur le serveur MDM .

Dans cet exemple, l'objectif est de charger une nouvelle agence dans l'entité métier *Agency*. Cette nouvelle agence doit comprendre un identifiant, un nom et une ville.

- A partir de la **Palette**, glissez les composants **tFixedFlowInput** de la famille **Misc**, et **tMDMOutput** de la famille **Talend MDM** dans l'espace de modélisation.
- Reliez-les via un lien de type **Row > Main**.
- Double-cliquez sur le composant **tFixedFlowInput** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.

- Dans la liste **Schema**, sélectionnez **Built-In** et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour décrire la structure des données maître que vous souhaitez écrire dans le serveur MDM.

- Ajoutez trois colonnes de type **String** en cliquant sur le bouton **[+]**. Nommez ces colonnes *Id*, *Name* et *City*.
- Cliquez sur **OK**pour valider vos modifications et passer à l'étape suivante.
- Dans le champ **Number of rows**, saisissez le nombre de lignes que vous souhaitez générer.
- Dans la zone **Mode**, sélectionnez le mode **Use Single Table** pour ne générer qu'une seule table.
- Dans le tableau **Values**, saisissez entre guillemets, dans chaque champ **Value**, la valeur correspondant à chaque colonne du schéma.
- Dans l'espace de modélisation, double-cliquez sur le composant **tMDMOutput** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.

- Dans la liste **Property Type**, sélectionnez **Built-In** et renseignez les champs manuellement. Si vous avez stocké les informations de connexion au MDM dans les métadonnées du Repository, sélectionnez **Repository** dans la liste et les champs suivants seront renseignés automatiquement.
- Dans la liste **Schema**, sélectionnez **Built-In** et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour décrire la structure des données maître que vous souhaitez charger sur le serveur MDM.

Le composant **tMDMOutput** génère un document XML, l'écrit dans un champ de sortie et l'envoie au serveur MDM, afin que le schéma de sortie ait toujours une colonne *xml* en lecture seule.

- Cliquez sur **OK** pour passer à l'étape suivante.

  La liste **Result of the XML serialization** dans l'onglet **Basic settings** est automatiquement renseignée avec la colonne *xml* de sortie que vous venez de créer.

- Dans le champ **URL**, saisissez entre guillemets l'URL de connexion au serveur MDM.
- Dans les champs **Username** et **Password**, saisissez l'identifiant et le mot de passe de connexion au serveur MDM.
- Dans le champ **Version** (déprécié), saisissez entre guillemets le nom de la Version des données maître à laquelle vous souhaitez accéder. Laissez le champ vide pour accéder à la Version par défaut.

- Dans le champ **Data Model**, saisissez entre guillemets le nom du modèle de données par rapport auquel vous souhaitez valider les données maître que vous souhaitez écrire.
- Dans le champ **Data Container**, saisissez entre guillemets le nom du conteneur de données dans lequel les données maître doivent être écrites.
- Cochez la case **Is Update** si vous souhaitez uniquement mettre à jour des champs et non l'enregistrement de données complet.
- Dans la vue **Component**, cliquez sur l'onglet **Advanced settings** pour paramétrer les paramètres avancés du composant.

- Cochez la case **Extended Output** si vous souhaitez commiter les données maître par lot. Vous pouvez spécifier le nombre de lignes par lot dans le champ **Rows to commit**.
- Cliquez sur le bouton **[...]** du champ **Configure Xml Tree** pour ouvrir l'éditeur du composant.

- Dans la zone **Link target** à droite, cliquez dans le champ **Xml Tree** et remplacez **rootTag** par le nom de l'entité métier dans laquelle vous souhaitez insérer l'enregistrement de données, *Agency* dans cet exemple.

- Dans la zone **Linker source**, sélectionnez les trois colonnes de votre schéma et glissez les sur le nœud **Agency**.

La boîte de dialogue **[Selection]** s'ouvre.

- Sélectionnez **Create as sub-element of target node** pour que vos trois colonnes correspondent à trois sous-éléments du nœud **Agency** en XML. Cliquez sur **OK**.

- Dans la zone **Link target** cliquez-droit sur l'élément que vous souhaitez définir comme élément sur lequel effectuer une boucle, puis sélectionnez **Set as Loop Element** dans le menu contextuel. Dans cet exemple, **City** est l'élément d'itération.

- Cliquez sur **OK** pour valider vos changements et fermer l'éditeur.

- Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

Le nouvel enregistrement de données est inséré dans l'entité métier *Agency*, dans le conteneur de données *DStar*, sur le serveur MDM. Cet enregistrement de données contient, comme défini dans le schéma, l'ID de l'agence, son nom et sa ville. La clé primaire de l'enregistrement inséré est également générée et retrouvée.

Data Container Browser DStar
From
To
Entity
Agency [HEAD]
Keys
Keywords
Use Full Text Search
Page 3 of 3
Number of lines per page: 20
Display items of 41-43 of 43
Date
Entity
Keys
20101102 17:19:17 Agency NV01
20101102 17:19:17 Agency UT01
20101123 17:03:23 Agency PA05
XML Editor/Viewer
Tree
Source
Agency
Id: PA05
Name: Talend
City: Suresnes
Data Model
DStar
Save
Cancel

tMDMReceive

Propriétés du tMDMReceive

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMReceive** reçoit un enregistrement MDM provenant de déclencheurs MDM ou de processus MDM.	
Objectif	Ce composant décode un paramètre de contexte comprenant des données MDM XML et les transforme en schéma plat.	
Basic settings	*Property Type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier du Repository dans lequel les propriétés sont stockées. Les champs qui suivent sont automatiquement renseignés via les données récupérées.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	XML Record	Saisissez le paramètre de contexte vous permettant de récupérer les dernières modifications apportées au serveur MDM. Pour plus d'informations à propos de la création et de l'utilisation des paramètres de contexte, consultez le *Guide utilisateur du Studio Talend*.
	XPath Prefix	Si nécessaire, sélectionnez dans la liste l'expression XPath de boucle, qui est la concaténation du préfixe et du chemin de la boucle. `/item` : sélectionnez ce préfixe XPath lorsque le composant reçoit l'enregistrement d'un processus, car les processus encapsulent l'enregistrement dans un élément "item" uniquement. /exchange/item : sélectionnez ce préfixe XPath lorsque le composant reçoit l'enregistrement d'un déclencheur, car les déclencheurs encapsulent l'enregistrement dans un élément ‘item" dans un élément "exchange".
	Loop XPath query	Nœud de l'arborescence XML sur lequel est basée la boucle.

	Mapping	**Column** : reflète le schéma tel qu'il est défini dans l'éditeur Edit Schema. **XPath query** : Saisissez les champs à extraire de la structure XML d'entrée. **Get Nodes** : Cochez cette case pour ajouter les balises XML entourant les données à récupérer.
	Limit	Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n'est lue ou traitée.
	Die on error	Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row** > **Reject**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de sortie. Il nécessite un flux de sortie.	

Scénario : Extraire des informations d'un enregistrement MDM au format XML

Le scénario suivant décrit un Job simple extrayant des informations d'un enregistrement MDM au format XML et affiche ces informations dans la console.

Prérequis du scénario

Un conteneur de données *Product* et un modèle de données *Product* sont créés et déployés dans le serveur MDM. Les entités de données *Product* et *Store* sont définies et des enregistrements de données existent déjà au sein de ces entités.

Les entités *Product* et *Store* sont liées par une clé étrangère, le nom du magasin (*Name* et *Store*).

Cet exemple est conçu pour obtenir des informations concernant un nouveau produit. Lorsque vous créez un enregistrement *Product*, assurez-vous que les informations *Store* sont bien ajoutées pour le nouvel enregistrement.

Les entités et leurs attributs sont affichés ci-dessous.

Pour plus d'informations concernant les principes de fonctionnement de MDM, consultez la partie MDM dans le *Guide utilisateur du Studio Talend*.

Construire le Job

1. Déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : un **tMDMReceive** et un **tLogRow**.
2. Reliez le **tMDMReceive** au **tLogRow** à l'aide d'un lien **Row** > **Main**.
3. Renommez les composants afin de mieux identifier leur rôle.

Configurer les composants

Définir une variable de contexte

1. Dans l'onglet **Contexts**, cliquez sur le bouton **[+]** pour ajouter une variable. Nommez-la *exchangeMessage*.

2. Saisissez la valeur de la variable dans le champ **Value**.

   Notez que l'enregistrement XML doit suivre un schéma particulier. Pour plus d'informations concernant le schéma, consultez la description des processus et schémas utilisés dans les processus MDM d'appels de Jobs, dans le *Guide utilisateur du Studio Talend*.

   Ci-dessous se trouve un échantillon de l'enregistrement XML de l'Update Report :

```
<exchange xmlns:mdm="java:com.amalto.core.plugin.base.xslt.MdmExtension">
<report>
<Update>
<UserName>administrator</UserName>
<Source>genericUI</Source>
<TimeInMillis>1381486872930</TimeInMillis>
<OperationType>ACTION</OperationType>
<RevisionID>null</RevisionID>
<DataCluster>Product</DataCluster>
<DataModel>Product</DataModel>
<Concept>Product</Concept>
<Key>2</Key>
</Update>
</report>
<item><Product><Id>001</Id><Name>Computer</Name><Description>Laptop
 series</Description><Availability>true</Availability><Price>400</
Price><OnlineStore>TalendShop@@http://www.cafepress.com/Talend.2</
OnlineStore><Stores><Store>[Dell]</Store><Store>[Lenovo]</Store></Stores></
Product></item>
</exchange>
```

   Dans cet exemple, les espaces blancs de l'enregistrement XML sont retirés :

```
<exchange><report/><item><Product><Id>001</Id><Name>Computer</
Name><Description>Laptop series</Description><Availability>true</
Availability><Price>400</Price><OnlineStore>TalendShop@@http://www.cafepress.com/
Talend.2</OnlineStore><Stores><Store>[Dell]</Store><Store>[Lenovo]</Store></
Stores></Product></item></exchange>
```

3. Appuyez sur les touches **Ctrl+S** afin de sauvegarder vos modifications.

Configurer le tMDMReceive et le tLogRow

1. Double-cliquez sur le **tMDMReceive** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir la structure des données. Dans cet exemple, trois colonnes sont ajoutées : *Product_ID*, *Product_Name* et *Store_Name*.

3. Dans le champ **XML Record**, saisissez la variable de contexte *context.exchangeMessage*.

4. Dans la liste **XPath Prefix**, sélectionnez *"/exchange/item"*.

5. Dans le champ **Loop XPath query**, saisissez le nom de la balise racine de l'arbre XML. Dans cet exemple, saisissez *"/Product/Stores/Store"*.

6. La colonne **Column** de la table **Mapping** est alimentée par les colonnes définies dans le schéma. Dans la colonne **XPath query**, saisissez votre requête XPath. Dans cet exemple, l'ID du produit, le nom du produit et le nom du magasin où se trouve le produit sont les informations extraites.

7. Double-cliquez sur le composant **tLogRow** pour ouvrir sa vue **Basic settings**.

8. Sélectionnez l'option **Table (print values in cells of a table)** dans la zone **Mode**.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Exécutez le Job en appuyant sur la touche **F6** ou en cliquant sur le bouton **Run** dans l'onglet **Run**.

   Les informations des produits sont extraites de l'enregistrement XML et affichées dans la console.

```
[statistics] connecting to socket on port 3838
[statistics] connected
.----------+------------+----------.
|           Product_Info           |
|=---------+------------+---------=|
|Product_ID|Product_Name|Store_Name|
|=---------+------------+---------=|
|001       |Computer    |[Dell]    |
|001       |Computer    |[Lenovo]  |
'----------+------------+----------'

[statistics] disconnected

Job Product_info ended at 17:48 17/10/2013. [exit
code=0]
```

tMDMRollback

Propriétés du tMDMRollback

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMRollback** rétablit l'état original de la base de données, avant l'exécution d'un Job, au lieu de commiter les modifications.	
Objectif	Ce composant est utilisé comme partie d'une transaction MDM afin d'annuler les modifications effectuées dans la base de données et de ne pas les commiter définitivement, notamment pour éviter les commits partiels si une erreur survient.	
Basic settings	*Component List*	Sélectionnez le composant **tMDMConnection** pour lequel effectuer l'action de rollback.
	Close Connection	Cochez cette case pour fermer la session de la connexion au serveur MDM après avoir annulé les modifications. Notez que, même si la case n'est pas cochée, la connexion ne peut toujours pas être utilisée par le sous-job suivant, sauf si le mode Auto-Commit est activé.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau des composants.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est généralement utilisé avec les composants **tMDMConnection**, **tMDMCommit**, **tMDMSP**, **tMDMViewSearch**, **tMDMInput**, **tMDMDelete**, **tMDMRouteRecord**, **tMDMOutput** et **tMDMClose**.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Supprimer des données maître d'un hub MDM*.

tMDMRouteRecord

Propriétés du tMDMRouteRecord

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMRouteRecord** soumet la clé primaire d'un enregistrement stocké dans votre hub MDM à l'**Event Manager**, afin que ce gestionnaire déclenche le ou les processus attendu(s) selon certaines conditions que vous devez définir dans les pages des processus et des déclencheurs de la perspective **MDM** du *Studio Talend*. Pour plus d'informations concernant l'**Event Manager** et les processus MDM, consultez le *Guide utilisateur du* Studio Talend.	
Objectif	Ce composant permet à l'**Event Manager** d'identifier les modifications apportées à vos données, pour que les actions corrélées puissent être déclenchées.	
Basic settings	*Use an existing connection*	Cochez cette case pour utiliser un composant **tMDMConnection** déjà configuré.
	URL	Saisissez l'URL d'accès au serveur MDM.
	Username et *Password*	Informations d'authentification de l'utilisateur du serveur MDM. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Version (déprécié)	Saisissez le nom de la Version des données maître à laquelle vous souhaitez vous connecter, en fonction de vos droits utilisateur. Laissez ce champ vide pour afficher la Version par défaut des données maître.
	Data Container	Saisissez le nom du conteneur de données contenant les données que vous souhaitez effacer.
	Type	Sélectionnez **Master** ou **Staging** pour spécifier sur quelle base de données effectuer l'action.
	Entity Name	Saisissez le nom de l'entité métier contenant l'enregistrement que vous souhaitez faire lire à l'**Event Manager**.
	IDs	Spécifiez la (les) clé(s) primaire(s) de(s) l'enregistrement(s) que vous souhaitez faire lire à l'**Event Manager**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Iterate. **Trigger** : Run If, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError. Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate. **Trigger** : Run If, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError. Pour plus d'informations concernant les connexions, consultez la section relative aux types de connexion, dans le *Guide utilisateur du Studio Talend*.	

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant de début. Il nécessite un flux de sortie.

Scénario : Router un enregistrement vers l'Event Manager

Dans ce scénario, le composant **tMDMRouteRecord** est utilisé pour soumettre à l'**Event Manager** la clé primaire d'un enregistrement notifiant une mise à jour, afin que l'**Event Manager** déclenche un processus informant l'utilisateur de la mise à jour effectuée.

Le Studio Talend est sensible à la casse, respectez donc les majuscules et les minuscules lorsque vous réalisez ce scénario.

Prérequis du scénario

Les prérequis pour la réalisation de ce scénario sont les suivants :

- Un conteneur de données contenant différents enregistrements utilisant un modèle spécifique. Dans ce scénario, le conteneur de données est *Product*, et un enregistrement est inclus dans le conteneur, en suivant le modèle nommé *Product* :

Cette capture d'écran présente l'un des enregistrements produit stockés avec tous les attributs que vous pouvez voir.

Pour plus d'informations concernant la création d'un conteneur de données et d'un modèle de données, consultez le *Guide utilisateur du* Studio Talend.

Pour plus d'informations concernant la création d'un enregistrement et l'accès à ses attributs, consultez le *Guide utilisateur de Talend MDM Web User Interface*.

Créer un message pour informer l'utilisateur qu'un enregistrement est mis à jour

Un Job, utilisé pour informer l'utilisateur de la mise à jour. Dans ce scénario, le Job est appelé *message*, il n'utilise que le composant **tMsgBox**.

- Déposez un **tMsgBox** de la **Palette** dans l'espace de modélisation graphique.
- Double-cliquez sur le composant afin d'afficher et de configurer ses **Basic settings** :

- Dans le champ **Title**, saisissez "*Talend MDM*".
- Dans le champ **Message**, saisissez le message qui apparaîtra : *"A record is updated"*.
- Sauvegarder votre Job et déployez-le sur le serveur MDM.

Pour plus d'informations concernant le composant **tMsgBox**, consultez *tMsgBox*.

Pour plus d'informations sur comment déployer un Job sur le serveur MDM, consultez le *Guide utilisateur du* Studio Talend.

Router un enregistrement afin de déclencher le processus correspondant

Ce scénario vous explique comment déclencher un processus à l'aide d'un composant **tMDMRouteRecord**.

- Connectez-vous à *Talend MDM Web User Interface* et cliquez sur **Master Data Browser**.

  Pour plus d'informations concernant la connexion à *Talend MDM Web User Interface* et l'ouverture de la vue **Master Data Browser**, consultez le *Guide utilisateur de Talend MDM Web User Interface*.

- Dans le coin supérieur droit de la page Web, cliquez sur le bouton afin d'afficher le panneau **Actions**.
- Dans le panneau **Actions** de droite, sélectionnez le conteneur de données requis, ainsi que le modèle de données contenant le fichier à mettre à jour. Dans ce scénario, le conteneur de données et le modèle de données sont tous les deux appelés *Product*.
- Cliquez sur **Save** afin de sauvegarder le conteneur et le modèle de données sélectionnés.
- Dans la vue **Master Data Browser**, sélectionnez l'entité *Product*.
- Double-cliquez sur l'un des enregistrements produit afin d'afficher ses attributs dans une nouvelle vue dédiée à ce produit. Par exemple, ouvrez le produit *Talend Mug* ayant l'Id unique *231035938*.

- Dans cette vue, modifiez la valeur de l'un des attributs. Par exemple, mettez à jour ce produit. Rendez-le disponible en cochant la case **Availability**.
- Cliquez sur **Save** afin de valider la mise à jour.
- Ouvrez votre *Studio Talend* et assurez-vous que vous êtes connecté au serveur MDM. Pour plus d'informations concernant le lancement du *Studio Talend* et sa connexion au serveur MDM, consultez le *Guide utilisateur du Studio Talend*.
- Dans la vue **MDM Repository**, sous le nœud **Job Designs**, cliquez-droit sur le Job *message*.
- Dans le menu contextuel, sélectionnez **Generate Talend Job Caller Process**, acceptez les options par défaut et cliquez sur **Generate**. Le processus utilisé pour appeler ce Job est généré et stocké dans **Event Management** > **Process**.
- Déployez votre nouveau processus sur le serveur MDM.
- Sous le nœud **Event Management**, cliquez-droit sur **Trigger**.
- Dans le menu contextuel, sélectionnez **New**.
- Dans l'assistant **[New Trigger]**, nommez le déclencheur *TriggerMessage*.

- Cliquez sur **OK** afin d'ouvrir la vue du nouveau déclencheur dans l'espace de travail de votre studio.
- Configurez le déclencheur afin qu'il lance le processus appelant le Job *message* une fois la mise à jour effectuée.

- Dans le champ **Description**, saisissez *Trigger that calls the Talend Job: message_0.1.war*.
- Dans le champ **Entity**, sélectionnez l'entité métier sur laquelle vous souhaitez déclencher le processus, ou saisissez son nom. Dans cet exemple, son nom est *Update*.
- Dans le champ **Service JNDI Name**, sélectionnez **callprocess** dans la liste déroulante.
- Dans le champ **Service Parameters**, cochez la case *CallJob_message*.
- Dans la zone **Trigger xPath Expressions**, cliquez sur le bouton sous le tableau, afin d'ajouter une ligne XPath.
- Dans la ligne nouvellement ajoutée, cliquez sur le bouton **[...]** pour ouvrir une boîte de dialogue dans laquelle vous pouvez sélectionner l'entité ou l'élément sur laquelle/lequel vous souhaitez définir des conditions. Dans cet exemple, son nom est *Update/OperationType*.

- Dans la colonne **Value**, saisissez une valeur pour cette ligne. Dans cet exemple, la valeur est *UPDATE*.
- Dans la colonne **Condition Id**, saisissez un identifiant unique pour la condition que vous souhaitez définir, par exemple *C1*.
- Dans la zone **Conditions**, saisissez la requête que vous souhaitez exécuter sur les données en utilisant l'ID de la condition ID *C1* défini plus tôt.
- Appuyez sur **Ctrl+S** pour sauvegarder le déclencheur et déployez-le sur le serveur MDM.
- Dans la vue **MDM Repository**, double-cliquez sur **Data Container** > **System** > **UpdateReport** pour ouvrir la vue **Data Container Browser UpdateReport**. Le rapport **UpdateReport** contient les traces complètes de toutes les actions de création, de mise à jour ou de suppression (*Create*, *Update* ou *Delete*) effectuées sur les données maître.

  Notez, que, si le conteneur de données *UpdateReport* n'est pas disponible, vous devez d'abord l'importer de votre serveur MDM. Pour plus d'informations concernant l'import d'objets systèmes depuis le serveur MDM dans votre référentiel local, consultez le *Guide utilisateur du Studio Talend*.

- A côté du champ **Entity** de cette vue, cliquez sur le bouton pour rechercher tous les enregistrements concernant les actions dans le rapport **UpdateReport**. Notez que l'entité **Update** ne signifie pas nécessairement que l'action enregistrée est une mise à jour, c'est simplement le nom de l'entité définie par le modèle de données du rapport **UpdateReport** pouvant contenir différentes actions comme *create*, *delete*, et *update*.
- Le dernier enregistrement correspond aux modifications apportées à l'enregistrement produit au début du scénario. La clé primaire de cet enregistrement est *genericUI.1283244014172*. Cet enregistrement sera routé vers l'**Event trigger**.
- Dans la perspective **Integration**, cliquez-droit sur **Job Designs**, dans la vue **Repository**. Dans le menu contextuel, sélectionnez **Create Job**.
- Un assistant s'ouvre. Dans le champ **Name**, saisissez *RouteRecord*, puis cliquez sur **Finish**.
- Déposez le composant **tMDMRouteRecord** de la **Palette** dans l'espace de modélisation graphique.
- Double-cliquez sur ce composant afin d'ouvrir sa vue **Component**.

- Dans le champ **URL**, saisissez l'adresse de votre serveur MDM, *http://localhost:8080/talend/TalendPort, dans cet exemple*.
- Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant et votre mot de passe de connexion.
- Dans le champ **Data Container** saisissez le nom du conteneur de données contenant l'enregistrement que vous souhaitez router vers l'**Event Manager**. Dans cet exemple, son nom est *UpdateReport*.
- Dans le champ **Entity Name**, saisissez le nom de l'entité à laquelle appartient l'enregistrement que vous souhaitez router. Dans cet exemple, son nom est *Update*.
- Dans la zone **IDS**, cliquez sur le bouton **[+]** sous le tableau, afin d'ajouter une ligne.
- Dans cette nouvelle ligne, renseignez la clé primaire de l'enregistrement devant être routé vers l'**Event manager**, c'est-à-dire *genericUI.1283244014172*, comme vous l'avez vu dans le rapport **Data Container Browser UpdateReport**.
- Appuyez sur **F6** pour exécuter le Job. L'**Event manager** appelle le processus devant exécuter le Job *message* et générer la boîte de dialogue informant l'utilisateur de la mise à jour effectuée.

Le composant **tMDMRouteRecord** est utilisé pour soumettre à l'**Event Manager** la clé primaire d'un enregistrement notifiant une mise à jour. L'**Event Manager** analyse le fichier et vérifie que les conditions définies dans la vue de configuration du déclencheur *Trigger's message* sont remplies, puis appelle le processus lançant le Job *Message*. Ce Job informe l'utilisateur de la mise à jour effectuée en faisant apparaître une fenêtre pop-up.

tMDMSP

Propriétés du tMDMSP

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMSP** appelle une procédure stockée du hub MDM.	
Objectif	**tMDMSP** permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.	
Basic settings	*Schema* et *Edit Schema*	Dans une procédure stockée, le schéma est un paramètre d'entrée. Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule.
	Use an existing connection	Cochez cette case et sélectionnez le composant **tMDMConnection** adéquat à partir de la liste **Component** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	URL	Saisissez l'URL du serveur MDM.
	Username et *Password*	Informations d'authentification sur l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Version (déprécié)	Saisissez le nom de la Version des données maître à laquelle vous souhaitez vous connecter, en fonction de vos droits utilisateur. Laissez ce champ vide pour afficher la Version par défaut des données maître.
	Data Container	Saisissez le nom du conteneur de données contenant les données que vous souhaitez effacer.
	Type	Sélectionnez **Master** ou **Staging** pour spécifier sur quelle base de données effectuer l'action.
	Procedure Name	Saisissez le nom exact de la Procédure Stockée.
	Parameters (in order)	Cliquez sur le bouton **[+]** et sélectionnez les colonnes d'entrée (**Input Columns**) requises pour les procédures. Le schéma SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Connections	Liens de sortie (d'un composant à l'autre) : **Row** : Main **Trigger** : Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError. Liens d'entrée (d'un composant à l'autre) : **Row** : Main, Iterate; Trigger : Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError	

	Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début mais aucun paramètre d'entrée n'est requis pour la procédure à appeler. Un lien de sortie est requis.

Scénario : Exécuter une procédure stockée dans le hub MDM

Le but du Job suivant est de calculer le prix total de chaque produit enregistré dans votre interface Web MDM.

Ce Job va générer des paramètres pour l'exécution d'une procédure stockée dans le hub MDM, puis extraire les données souhaitées du résultat retourné au format XML et présenter les données extraites dans le studio.

Les produits dont le prix est à traiter sont listés dans votre interface Web MDM.

Pour ce Job, vous devez avoir précédemment créé une procédure stockée nommée *PriceAddition* dans la vue **MDM Repository** et déployé cette procédure stockée sur le serveur. La procédure utilise les étapes suivantes :

```
for $d in distinct-values(//Product/Name)
let $product := //Product[Name= $d and Price >= %0 and Price <=%1]
```

```
order by $d
return <result><Name>{$d}</Name><Sum>{sum($product/Price)}</Sum></result>
```

Pour plus d'informations concernant comment travailler avec des procédures stockées, consultez le *Guide utilisateur du Studio Talend*.

Pour créer ce Job, procédez comme suit :

1. Glissez-déposez les composants suivants utilisés dans cet exemple : **tFixedFlowInput**, **tMDMSP**, **tExtractXMLField**, **tLogRow**.

2. Connectez les composants à l'aide de liens **Row** > **Main**.

3. Le composant **tFixedFlowInput** sert à générer la gamme de prix vous intéressant pour ce calcul. Dans cet exemple, définissez *10* comme minimum et *17* comme maximum afin de prendre en compte tous les produits. Pour commencer, double-cliquez sur le composant **tFixedFlowInput** afin d'ouvrir sa vue **Component**.

4. Dans la vue **Component**, cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin d'ouvrir l'éditeur de schéma de ce composant.

5. Dans l'éditeur de schéma, ajoutez les deux paramètres *min* et *max* utilisés pour définir la gamme de prix.

6. Cliquez sur **OK**.

7. Dans la table **Values** dans la zone **Mode** de la vue **Component**, les deux paramètres *min* et *max* que vous avez définis dans l'éditeur de schéma du composant s'affichent.

8. Dans la colonne **Value** de la table **Values**, saisissez le nombre *10* pour le paramètre *min* et *17* pour le paramètre *max*.

9. Double-cliquez sur le composant **tMDMSP** afin d'ouvrir sa vue **Component**.

10. Dans le champ **URL** de la vue **Component**, saisissez l'adresse du serveur MDM, dans cet exemple : *http://localhost:8080/talend/TalendPort*.

11. Dans les champs **Username** et **Password**, saisissez les données d'authentification, dans cet exemple : *admin* et *talend* respectivement.

12. Dans les champs **Data Container** et **Procedure Name**, saisissez les noms exacts du conteneur de données *Product* et de la procédure stockée *PriceAddition*.

13. Sous la table **Parameters (in order)**, cliquez deux fois sur le bouton **[+]** afin d'ajouter deux lignes à cette table.

    Dans la table **Parameters (in order)**, cliquez sur chaque ligne ajoutée et, dans la liste déroulante, sélectionnez le paramètre *min* pour l'une et le paramètre *max* pour l'autre.

14. Double-cliquez sur le composant **tExtractXMLField** afin d'ouvrir sa vue **Component**.

15. Dans la vue **Component**, cliquez sur le bouton **[...]** situé à côté du champ **Edit schema** afin d'ouvrir l'éditeur de schéma du composant.

16. Dans l'éditeur de schéma, ajoutez deux colonnes afin de définir la structure des données sortantes. Ces deux colonnes sont *name* et *sum*. Elles représentent respectivement le nom et le prix total de chaque type de produit enregistré dans l'interface Web.

17. Cliquez sur **OK** afin de valider la configuration. Les deux colonnes s'affichent dans la table **Mapping** de la vue **Component**.

18. Dans le champ **Loop XPath Query**, saisissez le nœud de l'arborescence XML sur lequel effectuer une boucle. Dans l'exemple, le nœud est */result* comme vous pouvez lire dans le code de la procédure : `return <result><Name>{$d}</Name><Sum>{sum($product/Price)}</Sum></result>`.

19. Dans le champ **XPath Query** de la table **Mapping**, saisissez le nom exact du nœud sur lequel effectuer la boucle. Pour la colonne *name*, saisissez "*/result/Name*", pour mapper les noms du produit. Pour la colonne *sum*, saisissez "*/result/Sum*", pour extraire les prix totaux.

20. Double-cliquez sur le composant **tLogRow** afin d'ouvrir sa vue **Component**.

21. Synchronisez le schéma avec celui du composant précédant (en cliquant sur le bouton **Sync columns**).

22. Dans la zone **Mode**, sélectionnez l'option **Table (print values in cells of a table)** pour une meilleure lecture.

23. Appuyez ensuite sur **F6** afin d'exécuter le Job.

    Les données sortantes s'affichent dans la console de la vue **Run**.

```
Starting job feature14026_tMDMSP at 17:29 25/08/2010.

[statistics] connecting to socket on port 3487
[statistics] connected
.-------------------------+-----------------.
|                tLogRow_1                  |
|=------------------------+----------------=|
|name                     |sum              |
|=------------------------+----------------=|
|Talend Golf Shirt        |16.99            |
|Talend Jr. Spaghetti Tank|16.99            |
|Talend Large Mug         |11.99            |
|Talend Mug               |94.92999999999999|
'-------------------------+-----------------'

[statistics] disconnected
Job feature14026_tMDMSP ended at 17:29 25/08/2010. [exit
code=0]
```

La sortie liste les quatre sortes de produit enregistrés dans l'interface Web MDM et le prix total pour chaque type de produit.

tMDMTriggerInput

Propriétés du tMDMTriggerInput

Famille de composant	Talend MDM	
Fonction	Une fois exécuté, le composant **tMDMTriggerInput** lit un message XML (de type **Document**) envoyé par MDM et le passe au composant suivant. Ce composant fonctionne avec le nouveau service de déclenchement et le plug-in de traitement en version 5.0 et supérieure de MDM. Les Jobs, les déclencheurs et les processus MDM développés dans les versions précédentes de MDM restent supportés. Il est cependant recommandé d'utiliser ce composant lors de la création de nouveau Jobs MDM.	
Objectif	A chaque sauvegarde d'une modification dans votre MDM, l'enregistrement modifié correspondant est généré au format XML. Lors de l'exécution, ce composant lit l'enregistrement et envoie les informations relatives au composant suivant. Avec ce composant, il n'est pas nécessaire de configurer votre Job afin de le faire communiquer les modifications de données de MDM vers votre Job.	
Basic settings	*Property Type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement. Le **tMDMTriggerInput** doit utiliser cette option pour appliquer le schéma en lecture seule par défaut. La colonne **MDM_message** est l'unique colonne de ce schéma.
		Repository : Sélectionnez le fichier du Repository dans lequel les propriétés sont stockées. Les champs qui suivent sont automatiquement renseignés via les données récupérées. Comme le composant **tMDMTriggerInput** fournit un schéma fixe en lecture seule, vous devez utiliser l'option **Built-in**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

		C'est l'option par défaut pour le **tMDMTriggerInput**. Avec cette option, le schéma en lecture seule est utilisé pour traiter le schéma du message MDM au format XML.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*. Comme le composant **tMDMTriggerInput** fournit un schéma fixe en lecture seule, vous devez utiliser l'option **Built-in**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de début. Il requiert un flux de sortie. Pour recevoir le message de MDM, vous devez déployer le Job à l'aide de ce composant sur votre serveur MDM et générer le déclencheur et le processus correspondants dans MDM pour invoquer ce Job. Pour plus d'informations concernant le déploiement d'un Job sur le serveur MDM et la génération d'un déclencheur ou d'un processus, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations concernant la modification d'un enregistrement dans MDM, consultez le *Guide utilisateur de Talend MDM Web User Interface*.	
Limitation	Durant le déploiement de ce composant sur le serveur MDM, vous devez sélectionner le type **Hosted (Zip)** comme format du Job déployé. Si vous le déployez dans le type **Distributed (War)**, le Job relatif ne peut être invoqué. Pour plus d'informations concernant les types disponibles, consultez le *Guide utilisateur du Studio Talend*.	

Scénario : Échanger des informations concernant les événements d'un enregistrement MDM

Dans ce scénario, un Job à quatre composants est utilisé pour échanger des informations sur des événements concernant un enregistrement produit. A l'aide d'une connexion MDM établie depuis le **Repository**, ce Job est déclenché par le *Studio Talend*, une fois que vous avez mis à jour un enregistrement produit mis à jour.

Pour reproduire ce scénario, accomplissez les tâches suivantes, dans l'ordre :

1. Créez une connexion MDM de type **Receive** dans le **Repository** du Studio. Cette connexion au hub MDM contient les enregistrements que vous souhaitez mettre à jour.

2. Créez le Job recevant et envoyant le message MDM mis à jour.

3. Générer le processus invoquant ce Job.

4. Mettre à jour un enregistrement MDM spécifique.

Pour créer les enregistrements, les modèles et les conteneurs MDM utilisés dans ce scénario, vous pouvez exécuter les Jobs du projet démo MDM dans le *Studio Talend* puis mettre à jour le serveur MDM afin de déployer les objets créés et qu'ils soient pris en compte lors de l'exécution. Vous allez utiliser ce serveur dans tout le scénario.

Pour plus d'informations concernant l'import d'un projet démo, consultez le *Guide utilisateur du Studio Talend*.

Pour plus d'informations concernant la mise à jour du serveur pour déployer des objets, consultez le *Guide utilisateur du Studio Talend*.

Pour plus d'informations concernant les événements et la gestion des événements MDM, consultez le *Guide utilisateur du Studio Talend*.

Créer une connexion MDM

Etablir la connexion

1. Lancez le serveur MDM auquel vous devez communiquer le message mis à jour.
2. Dans la perspective **Integration** du Studio, développez le nœud **Metadata** dans le **Repository**.
3. Cliquez-droit sur l'élément **Talend MDM** et sélectionnez **Create MDM connection**.

4. Saisissez, dans le champ **Name**, le nom que vous souhaitez utiliser pour cette connexion, et, si nécessaire, ajoutez un objectif dans le champ **Purpose** et une **Description**. Par exemple, nommez cette connexion *receive_update*.
5. Dans l'étape suivante, saisissez les informations d'authentification utilisées pour vous connecter au service Web MDM via lequel vous gérez l'enregistrement à mettre à jour.

Une fois que vous avez cliqué sur le bouton **Check** et que la connexion est établie, vous pouvez cliquer sur le bouton **Next**, qui devient accessible.

6. Dans l'étape suivante, sélectionnez la **Version** (déprécié), le modèle de données (**Data model**) et le conteneur de données (**Data Container**) utilisés par l'enregistrement à mettre à jour. Dans ce scénario, le modèle et le conteneur se nomment *Product*.

7. Cliquez sur **Finish** pour valider la création. La connexion créée apparaît sous le nœud **Metadata** dans le **Repository**.

Récupérer les entités

1. Cliquez-droit sur la connexion créée et, dans le menu contextuel, sélectionnez **Retrieve entities**. L'assistant apparaît.

2. Sélectionnez **Receive MDM** et cliquez sur **Next** pour continuer.

3. Sélectionnez l'entité à récupérer. Dans ce scénario, l'entité se nomme *Product*. Le nom est automatiquement renseigné.

4. Dans l'étape suivante, déposez les éléments à récupérer de la zone **Source Schema** dans la zone **Target Schema**. Dans ce scénario, l'élément *Features* est l'élément de boucle et les éléments *Id*, *Name* et *Description* sont les champs à extraire.

5. Dans l'étape suivante, si nécessaire, modifiez la description du schéma récupéré ; sinon, cliquez sur **Finish** afin de finaliser la récupération de l'entité. Dans ce scénario, laissez la description par défaut du schéma et cliquez sur **Finish**.

   Le schéma de l'entité est récupéré. Pour plus d'informations concernant le conteneur et le modèle de données utilisés par la solution MDM de Talend, consultez le *Guide utilisateur du Studio Talend*.

Créer le Job communiquant le message MDM

Relier les composants

1. Dans la perspective **Integration** du Studio, cliquez-droit sur **Job Design** dans le **Repository** et sélectionnez **Create job**. L'assistant **[New Job]** apparaît.

2. Donnez un nom au Job et cliquez sur **Finish** pour fermer l'assistant et valider la création. Un Job vide s'ouvre dans l'espace de modélisation graphique du Studio.

3. Déposez un **tMDMTriggerInput**, un **tXMLMap**, un **tMDMTriggerOutput** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.

4. Cliquez-droit sur le **tMDMTriggerInput** et, dans le menu contextuel, sélectionnez le lien **Row** > **Main** afin de le connecter au **tXMLMap**.

5. Répétez l'opération pour relier le **tXMLMap** au **tMDMTriggerOutput**. Une boîte de dialogue apparaît et vous demande de nommer le lien créé.

6. Double-cliquez sur le **tMDMTriggerOutput** pour ouvrir sa vue **Component**.

7. Cliquez sur **Edit schema** pour ouvrir l'éditeur.

8. Sélectionnez la colonne prédéfinie du **tMDMTriggerOutput**, cliquez sur le bouton afin de propager cette colonne dans le schéma d'entrée (à gauche).

Configurer la transformation du message MDM

1. Double-cliquez sur le **tXMLMap** pour ouvrir son éditeur.

2. Dans la table représentant le flux d'entrée, cliquez-droit sur la colonne **MDM_Message**, en haut de l'arborescence XML et sélectionnez **Import from repository**. L'assistant **[Metadata]** apparaît.

3. Sélectionnez le schéma de l'entité précédemment récupéré à l'aide du modèle **Receive MDM** puis cliquez sur OK. Dans ce scénario, le schéma de l'entité est *ProductReceive*.

4. Une boîte de dialogue apparaît et vous propose d'ajouter le schéma du rapport **Update Report** dans l'arborescence XML d'entrée. Cliquez sur **OK** pour accepter. Cela permet de construire un document complet d'entrée pour un événement MDM. Dans l'arborescence XML d'entrée, l'élément *Features* est automatiquement défini comme élément de boucle.

5. Dans la table représentant le flux de sortie, développez l'arbre XML de sortie, comme dans la capture d'écran ci-dessous. Cette arborescence est construite selon le modèle statique requis du rapport de sortie MDM.

Le document XML attendu pour le message de confirmation de validation est

```
<report><message type="info">message</message></report>
```

Le document XML attendu pour le message d'échec de validation est

```
<report><message type="error">message</message></report>
```

6. Mappez l'élément *OperationType* d'entrée à l'élément *message* de sortie. Cela permet d'écrire en sortie les informations concernant le type d'événement se déroulant dans l'enregistrement MDM.

   Pour obtenir plus d'informations, vous pouvez concaténer plusieurs des éléments d'entrée que vous devez extraire, dans la colonne **Expression** de cet élément de message. Le **tMap** et le **tXMLMap** vous permettent d'éditer des expressions via l'éditeur d'expressions. Pour plus d'informations concernant l'édition d'une expression, consultez le *Guide utilisateur du Studio Talend*.

7. Dans la colonne **Expression**, saisissez `"info"` dans la ligne correspondant à *@type*.

8. Cliquez sur l'icône de clé anglaise pour afficher le panneau des paramètre, puis configurez l'option **All in one** à **true**.

9. Cliquez sur **OK** pour fermer l'éditeur et valider les modifications.

10. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Component**, puis cliquez sur **Sync columns**.

Le Job est terminé. Pour plus d'informations concernant about le document d'entrée et le rapport de sortie d'un événement MDM, consultez le *Guide utilisateur du Studio Talend*.

Générer le processus invoquant le Job créé

Déployer le Job à appeler sur le serveur MDM

1. Passez en perspective **MDM** en cliquant sur le bouton correspondant, dans le coin supérieur droit du Studio.

2. Dans la vue **MDM Repository**, cliquez sur le bouton **Refresh** afin que le Job créé apparaisse sous le nœud **Job Designs** du **Repository**.

3. Cliquez-droit sur le Job, *update_product* dans ce scénario et, dans le menu contextuel, sélectionnez **Deploy to** pour le déployer sur le serveur MDM.

4. L'assistant de déploiement apparaît. Dans la liste des serveurs, sélectionnez le serveur MDM que vous utilisez puis cliquez sur **OK**.

5. Dans la fenêtre **[Deploy to Talend MDM]** qui s'ouvre, sélectionnez un type d'export dans la liste **Export type** et cochez la case **Context scripts** pour le Job à déployer. Dans ce scénario, laissez les paramètres par défaut : **Export type** à **Hosted (zip)** et **Context scripts** à **Default**.

   Pour plus d'informations concernant ces paramètres, consultez le *Guide utilisateur du Studio Talend*.

6. Cliquez sur **Finish** pour valider les paramètres et commencer le déploiement. Lorsque le déploiement est terminé, un message s'ouvre pour indiquer que le déploiement a bien été effectué.

7. Cliquez sur **OK** pour fermer ce message. Une fenêtre s'ouvre et liste les objets déployés. Dans ce scénario, le Job *update_product* est déployé.

8. Cliquez sur **OK** pour terminer la procédure de déploiement.

Générer le processus utilisé pour appeler le Job

1. Cliquez-droit à nouveau sur le Job *update_product* et sélectionnez **Generate Talend Job Caller Process** dans le menu contextuel.

2. Dans la fenêtre qui s'ouvre, laissez les paramètres par défaut pour ce scénario : **Integrated** et **Embedded**. Pour plus d'informations concernant les options disponibles dans cette fenêtre, consultez le *Guide utilisateur du Studio Talend*.

3. Cliquez sur **Generate** pour débuter la génération. Une fois terminée, un processus nommé *CallJob_update_product* apparaît sous le nœud **Process** dans le **MDM Repository**.

4. Cliquez-droit sur ce processus, puis sélectionnez **Deploy to** dans le menu contextuel afin de le déployer sur le serveur MDM.

5. Dans l'assistant, sélectionnez le serveur que vous utilisez, puis cliquez sur **OK** pour ouvrir la fenêtre listant les objets déployés.

6. Cliquez sur **OK** pour fermer cette fenêtre et finaliser le déploiement. Le point d'interrogation disparaît de l'icône du processus.

7. Dans la vue **MDM Repository**, cliquez-droit sur le processus *CallJob_update_prodcut*, puis sélectionnez **Rename** dans le menu contextuel.

8. Dans la fenêtre, renommez le processus *beforeSaving_update_product*, selon le modèle de nommage requis. Cliquez sur **OK** pour valider.

9. Déployez à nouveau ce processus, comme décrit précédemment.

Mettre à jour un enregistrement produit

1. Connectez-vous au servie Web du hub MDM que vous utilisez.

2. Dans le panneau **Actions**, à droite, vérifiez que les **Data Container** et **Data Model** que vous utilisez sont bien nommés *Product*.

3. Dans la page **Data Browser**, lancez la recherche dans les entités produit afin de lister tous les enregistrements produits disponibles.

4. Sélectionnez l'enregistrement produit que vous souhaitez mettre à jour, par exemple *Talend Trucker Hat*. Les détails de cet enregistrement apparaissent dans la vue *Product*.

5. Mettez à jour l'un des attributs. Par exemple, modifiez le prix en *11.00*, puis cliquez sur **Save**.

   Le message concernant le type d'opération de cet événement a été envoyé au serveur MDM et, grâce au composant **tLogRow**, ce message est affiché dans la fenêtre du serveur MDM.

```
10:33:21,244 INFO  [STDOUT] <?xml version="1.0" encoding="UTF-8"?>
<root><report><message type="info">UPDATE</message></report></root>
```

Pour plus d'informations concernant l'utilisation du service Web MDM, consultez le *Guide utilisateur de Talend MDM Web User Interface*

tMDMTriggerOutput

Propriétés du tMDMTriggerOutput

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMTriggerOutput** reçoit un flux XML (de type **Document**) du composant précédent. Ce composant fonctionne avec le nouveau service de déclenchement et le plug-in de traitement en version 5.0 et supérieure de MDM. Les Jobs, les déclencheurs et les processus MDM développés dans les versions précédentes de MDM restent supportés. Il est cependant recommandé d'utiliser ce composant lors de la création de nouveau Jobs MDM.	
Objectif	Ce composant reçoit un flux XML afin de configurer le message MDM pour que MDM récupère le message lors de l'exécution. Avec ce composant, il n'est pas nécessaire de configurer votre Job afin de le faire communiquer les modifications de données de MDM vers votre Job.	
Basic settings	*Property Type*	Peut être **Built-in** ou **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Propriétés utilisées ponctuellement. Le composant **tMDMTriggerOutput** doit utiliser cette option pour appliquer le schéma en lecture seule par défaut. La colonne **MDM_message** est l'unique colonne de ce schéma.
		Repository : Sélectionnez le fichier du Repository dans lequel les propriétés sont stockées. Les champs qui suivent sont automatiquement renseignés via les données récupérées. Comme le **tMDMTriggerOutput** fournit un schéma fixe en lecture seule, vous devez utiliser l'option **Built-in**.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le **Repository**. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*. C'est l'option par défaut pour le **tMDMTriggerOutput**. Avec cette option, le schéma en lecture seule est utilisé pour traiter le schéma du message MDM au format XML.

		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*. Comme le composant **tMDMTriggerOutput** fournit un schéma fixe en lecture seule, vous devez utiliser l'option **Built-in**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant est un composant de fin. Il requiert un flux d'entrée. Pour envoyer un message au MDM, vous devez déployer le Job à l'aide de ce composant sur votre serveur MDM et générer le déclencheur et le processus correspondants afin d'invoquer ce Job dans MDM. Pour plus d'informations concernant le déploiement d'un Job sur le serveur MDM et la génération d'un déclencheur ou d'un processus, consultez le *Guide utilisateur du Studio Talend*.	
Limitation	Durant le déploiement de ce composant sur le serveur MDM, vous devez sélectionner le type **Hosted (Zip)** comme format du Job déployé. Si vous le déployez dans le type **Distributed (War)**, le Job relatif ne peut être invoqué. Pour plus d'informations concernant les types disponibles, consultez le *Guide utilisateur du Studio Talend*.	

Scénario associé

Pour un scénario associé, consultez *Scénario : Échanger des informations concernant les événements d'un enregistrement MDM*.

tMDMViewSearch

Propriétés du tMDMViewSearch

Famille de composant	Talend MDM	
Fonction	Le composant **tMDMViewSearch** sélectionne des enregistrements d'un hub MDM en appliquant les filtres que vous avez créés dans un but précis. Les données récupérées ont une structure XML. Pour plus d'informations concernant la vue permettant de définir des critères de filtres, consultez le *Guide utilisateur du Studio Talend*.	
Objectif	**tMDMViewSearch** permet de retrouver des enregistrements MDM d'un hub MDM.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** afin de récupérer le schéma du composant précédent.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur duStudio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.
	XML Field	Sélectionnez le nom de la colonne dans laquelle vous souhaitez écrire les données XML.
	Use an existing connection	Cochez cette case et sélectionnez le composant **tMDMConnection** adéquat à partir de la liste **Component** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.
	URL	Saisissez l'URL du serveur MDM.
	Username et *Password*	Informations d'authentification sur l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles, puis cliquez sur **OK** afin de sauvegarder les paramètres.
	Version (déprécié)	Saisissez le nom de la Version des données maître à laquelle vous souhaitez vous connecter, en fonction de vos droits utilisateur.

		Laissez ce champ vide pour afficher la Version par défaut des données maître.
	Data Container	Saisissez le nom du conteneur de données contenant les données que vous souhaitez effacer.
	Type	Sélectionnez **Master** ou **Staging** pour spécifier sur quelle base de données effectuer l'action.
	View Name	Saisissez le nom de la vue dont les filtres seront appliqués pour traiter les enregistrements.
	Operations	Remplissez ce tableau afin de créer une clause WHERE. Les paramètres à définir sont : - **XPath** : définissez le chemin de l'expression afin de sélectionner le nœud XML sur lequel appliquer les filtres. - **Functions** : sélectionnez un opérateur dans la liste déroulante, comme **Contains**, **Starts with**, **Equals**, etc. - **Value** : saisissez la valeur que vous souhaitez récupérer. La valeur peut être une chaîne de caractères (par (exemple "*Apple*", un élément Entité (par exemple "*Product/Name*") ou une variable personnalisée (par exemple "`$user_context.language`" ou "`$user_context/properties['location']}`"). - **Predicate** : sélectionnez le prédicat permettant de combiner les conditions de filtres de différentes manières. Le prédicat peut être **none**, **or**, **and**, **exactly**, etc. *Les paramètres sont sensibles à la casse.*
	Order (One Row)	Remplissez ce tableau afin de choisir l'ordre de présentation des enregistrements récupérés. Les paramètres à définir sont : - **XPath** : définissez le chemin de l'expression afin de sélectionner le nœud XML sur lequel appliquer le tri. - **Order** : sélectionnez l'ordre de présentation, soit **asc** (ascendant) ou **desc** (descendant). *Les paramètres sont sensibles à la casse.* *Pour le moment, seule la première ligne créée dans le tableau **Order** est valide.*
	Spell Threshold	Configurez cette option sur -1 afin de désactiver le seuil. Ce seuil est utilisé pour décider du niveau de tolérance des erreurs orthographiques.
	Skip Rows	Renseignez le nombre de lignes à ignorer afin de spécifier à partir de quelle ligne doit commencer le processus. Par exemple, si vous saisissez *8*, le processus commencera à la neuvième ligne.
	Max Rows	Saisissez le nombre maximal de lignes à traiter. Si le champ **Limit** est paramétré sur *0*, aucune ligne ne sera lue ou traitée. Par défaut, la valeur de ce champ est *-1*, ce qui signifie qu'aucune limite n'est définie.
Advanced settings	*Batch Size*	Nombre de lignes dans chaque lot traité. Par défaut, la valeur de ce champ est *-1*, ce qui signifie que toutes les lignes sont traitées dans un seul lot.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.
Utilisation	Utilisez ce composant pour récupérer des enregistrements spécifiques.	
Connections	Liens de sortie (de ce composant à un autre) : **Row** : Iterate **Trigger** : Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError.	

	Liens d'entrée (d'un autre composant à celui-ci) : **Row** : Iterate; **Trigger** : Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError. Pour plus d'informations concernant les connexions, consultez la section relative aux types de connexion, dans le *Guide utilisateur du Studio Talend*.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

Scénario : Récupérer des enregistrements d'un hub MDM via une vue existante

Ce scénario décrit un Job comprenant deux composants récupérant un enregistrement de données ayant une structure XML.

Dans cet exemple, vous allez sélectionner les informations concernant le T-shirt de l'entité *Product* via la vue *Browse_items_Product* créée dans le *Studio Talend*. Chaque enregistrement de l'entité contient les détails définis comme critères de filtre : *Id*, *Name*, *Description* et *Price* (ID, Nom, Description et Prix).

- A partir de la **Palette**, déposez un composant **tMDMViewSearch** et un **tLogRow** dans l'espace de modélisation graphique.
- Reliez les composants à l'aide d'un lien **Row** > **Main**.
- Double-cliquez sur le **tMDMViewSearch** afin d'ouvrir sa vue **Basic settings**, dans l'onglet **Component** et de définir les propriétés du composant.

- Dans la liste **Schema**, sélectionnez **Built-In** puis cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir une boîte de dialogue dans laquelle vous pouvez définir la structure des données XML que vous souhaitez écrire.

- Cliquez sur le bouton **[+]** pour ajouter une colonne de type **String**. Nommez-la *Tshirt*.
- Cliquez sur **OK** pour valider la modification et passer à l'étape suivante.
- Dans le champ **XML Field**, sélectionnez *Tshirt* comme colonne dans laquelle vous souhaitez écrire les données récupérées.

- Dans le champ **URL**, saisissez l'adresse de votre serveur MDM et renseignez respectivement votre identifiant et votre mot de passe de connexion dans les champs **Username** et **Password**. Dans cet exemple, utilisez l'URL par défaut, puis saisissez *admin* comme identifiant et mot de passe.
- Dans le champ **Data Container**, saisissez le nom du conteneur de données : *Product*.
- Dans le champ **View Name**, saisissez le nom de la vue : *Browse_item_Product*.
- Sous le tableau **Operations**, cliquez sur le bouton **[+]** pour ajouter une ligne.
- Dans le tableau **Operations**, définissez la colonne **XPath** comme *Product/Name*, ce qui signifie que l'opération de filtre sera effectuée sur le noeud *Name*. Sélectionnez ensuite *Contains* dans la colonne **Function** puis saisissez *Tshirt* dans la colonne **Value**.
- Sous le tableau **Order (One Row)**, cliquez sur le bouton **[+]** afin d'ajouter une ligne.
- Dans le tableau **Order (One Row)**, définissez la colonne **XPath** comme *Product/Id* et sélectionnez l'ordre **asc** dans la colonne **Order**.
- Dans l'espace de modélisation graphique, cliquez sur le **tLogRow** afin d'ouvrir sa vue **Basic settings** et définir ses propriétés.

- A côté du bouton **[...]** utilisé pour éditer le schéma, cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.
- Appuyez sur **F6** afin d'exécuter le Job.

```
Starting job feature14031_tMDMViewSearch at 17:37 10/08/2010.

[statistics] connecting to socket on port 3920
[statistics] connected
<result>
    <Id>1</Id>
    <Name>Tshirt-V</Name>
    <Description>green</Description>
    <Price>16.00 </Price>
</result>
<result>
    <Id>2</Id>
    <Name>Tshirt-G</Name>
    <Description>Green</Description>
    <Price>17.00</Price>
</result>
[statistics] disconnected
Job feature14031_tMDMViewSearch ended at 17:37 10/08/2010.
[exit code=0]
```

Dans la console de la vue **Run**, vous pouvez voir les enregistrements *Tshirt* retournés, avec leur structure XML, triés en ordre ascendant.

Composants Technical

Ce chapitre fournit des informations détaillées concernant les composants de la famille **Technical** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

Les composants Technical sont des composants orientés Java et effectuent des actions très techniques, comme le chargement de données dans la mémoire (en de petits sous-ensembles d'informations) et leur conservation pour un usage ultérieur lors du traitement.

tHashInput

Propriétés du tHashInput

Les composants de la famille Technical sont normalement masqués par défaut, dans la **Palette**. Pour savoir comment les afficher dans la **Palette**, consultez le *Guide utilisateur du Studio Talend*.

Famille de composant	Technical	
Fonction	Le composant **tHashInput** lit des données dans la mémoire cache, écrites par le **tHashOutput** afin de donner une grande vitesse aux flux de données.	
Objectif	Ce composant lit des données dans la mémoire cache, écrites par le composant **tHashOutput** afin de peupler rapidement de données et de faciliter les transactions comprenant un large volume de données.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de ligne, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant (Repository). Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Link with a tHashOutput	Cochez cette case pour connecter votre composant à un **tHashOutput**. Cette case est cochée par défaut.
	Component list	Liste déroulante comprenant les composants **tHashOutput** disponibles.
	Clear cache after reading	Cochez cette case pour vider le cache après lecture des données chargées par un composant **tHashOutput**. Ainsi, les composants **tHashInput** suivants, s'il y en a, ne pourront lire les données du cache chargées par le **tHashOutput**.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.	

	Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Ce composant est utilisé avec le **tHashOutput**. Il lit des données depuis la mémoire cache, écrites par le **tHashOutput**. Ensemble, ces deux composants permettent un accès très rapide aux données afin de faciliter les transactions comprenant un grand nombre de données.

Scénario 1 : Lire des données directement dans la mémoire cache afin d'y accéder rapidement

Le Job suivant lit dans la mémoire cache un large volume de données écrites par deux composants **tHashOutput** et envoie le contenu dans un **tFileOutputDelimited**. L'objectif de ce Job est de montrer la vitesse de lecture et d'écriture des données. En pratique, les données ainsi générées peuvent être utilisées comme table de référence dans certains cas où un grand volume de données doit être référencé.

Déposer et relier les composants

1. Glissez-déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : deux **tFixedFlowInput**, deux **tHashOutput**, un **tHashInput** et un **tFileOutputDelimited**.

2. Reliez le premier **tFixedFlowInput** au premier **tHashOutput** à l'aide d'un lien **Row** > **Main**.

3. Reliez le second **tFixedFlowInput** au second **tHashOutput** à l'aide d'un lien **Row** > **Main**.

4. Connectez le premier sous-job (composant **tFixedFlowInput_1**) au deuxième sous-job (composant **tFixedFlowInput_2**) à l'aide d'un lien **OnSubjobOk**.

5. Reliez le **tHashInput** au **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main**.

6. Connectez le deuxième sous-job au dernier sous-job à l'aide d'un lien **OnSubjobOk**.

Configurer les composants

Configurer les entrées de données et le cache

1. Double-cliquez sur le premier **tFixedFlowInput** pour afficher sa vue **Basic settings** et configurer ses propriétés de base.

2. Dans la liste **Schema**, sélectionnez **Built-In**.

   Vous pouvez sélectionner **Repository** dans la liste **Schema** afin de renseigner automatiquement les champs, si la métadonnée correspondante est stockée dans le **Repository**. Pour plus d'informations concernant la centralisation des métadonnées, consultez le chapitre relatif aux métadonnées dans le *Guide utilisateur du* Studio Talend.

3. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** afin de définir la structure des données du flux d'entrée. Dans ce cas, le schéma contient deux colonnes : *ID* et *ID_Insurance*. Cliquez sur **OK** pour fermer la boîte de dialogue.

4. Dans le champ **Number of rows**, saisissez le nombre d'entrées à écrire, ici *50000*.

5. Sélectionnez l'option **Use Single Table**. Dans la table **Values**, saisissez, dans la colonne **Value**, une valeur pour chaque colonne, ici *1* pour *ID* et *3* pour *ID_Insurance*.

6. Effectuez la même opération pour le second **tFixedFlowInput** et saisissez *2* pour *ID* et *4* pour *ID_Insurance*.

7. Double-cliquez sur le premier **tHashOutput** afin d'afficher sa vue **Basic settings** et définir ses propriétés.

8. Dans la liste **Schema**, sélectionnez **Built-In** puis cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent. Sélectionnez **Keep all** dans la liste **Keys management** et laissez la case **Append** cochée.

9. Effectuez la même opération pour le second **tHashOutput**. Cochez sa case **Link with a tHashOutput**

Configurer la récupération des données du cache et la sortie des données.

1. Double-cliquez sur le composant **tHashInput** pour afficher sa vue **Basic settings** et définir ses propriétés.

2. Dans la liste **Schema**, sélectionnez **Built-In**. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour définir la structure des données, la même que dans le **tHashOutput**.

3. Sélectionnez **tHashOutput_1** dans la liste **Component list**.

4. Double-cliquez sur le composant **tFileOutputDelimited** pour afficher sa vue **Basic settings** et configurer ses propriétés.

5. Dans la liste **Property Type**, sélectionnez **Built-In**. Dans le champ **File Name**, saisissez le chemin d'accès à votre fichier ou cliquez sur le bouton **[...]** afin de parcourir votre système jusqu'à votre fichier. Ici, le chemin d'accès est le suivant : *"E:/Allr70207V5.0/Talend-All-r70207-V5.0.0NB/workspace/out.csv"*

6. Cochez la case **Include Header** puis cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.

Exécuter le Job

Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

Vous pouvez voir que les nombreuses entrées ont été écrites et lues de manière très rapide.

Scénario 2 : Vider la mémoire avant d'y charger les données si une boucle existe dans le même sous-job

Ce scénario montre l'utilisation de l'option **Append** du composant **tHashOutput**, qui permet de supprimer des données répétitives ou indésirables s'il y a une boucle dans le sous-job du **tHashOutput**.

Pour construire le Job, procédez comme suit :

Déposer et relier les composants

1. Glissez-déposez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tLoop**, **tFixedFlowInput**, **tHashOutput**, **tHashInput** et **tLogRow**.

2. Reliez le **tLoop** au**tFixedFlowInput** à l'aide d'un lien **Row** > **Iterate**.

3. Connectez le composant **tFixedFlowInput** au **tHashOutput** à l'aide d'un lien **Row** > **Main**.

4. Reliez le **tHashInput** au **tLogRow** à l'aide d'un lien **Row** > **Main**.

5. Connectez le **tLoop** au **tHashInput** à l'aide d'un lien **OnSubjobOk**.

Configurer les composants

Configurer les données d'entrée et le cache

1. Double-cliquez sur le **tLoop** pour afficher sa vue **Basic settings**.

2. Dans la zone **Loop Type**, sélectionnez **For** comme type de boucle. Saisissez respectivement *1*, *2* et *1* dans les champs **From**, **To** et **Step**. Laissez cochée la case **Values are increasing**.

3. Double-cliquez sur le composant **tFixedFlowInput** pour afficher sa vue **Basic settings**.

4. Dans la liste **Schema**, sélectionnez **Built-In**.

Vous pouvez sélectionner **Repository** dans la liste **Schema** afin de renseigner automatiquement tous les champs si la métadonnée correspondante est stockée dans le **Repository**. Pour plus d'informations concernant la centralisation des métadonnées sous le nœud **Metadata**, consultez le *Guide utilisateur du* Studio Talend.

5. Cliquez sur **Edit schema** pour définir la structure des données du flux d'entrée. Dans ce scénario, le schéma contient une colonne *Name*.

6. Cliquez sur **OK** pour fermer la boîte de dialogue.

7. Renseignez le champ **Number of rows** afin de spécifier les entrées à écrire, par exemple *1*.

8. Cochez la case **Use Single Table**. Dans la table **Values**, assignez une valeur au champ **Name**, par exemple *Marx*.

9. Double-cliquez sur le **tHashOutput** pour affiche sa vue **Basic settings**.

10. Dans la liste **Schema**, sélectionnez **Built-In** et cliquez sur **Sync columns** pour récupérer le schéma du composant précédent. Sélectionnez **Keep all** dans la liste **Keys management** et décochez la case **Append**.

Configurer la récupération des données du cache et la sortie des données

1. Double-cliquez sur le **tHashInput** pour afficher sa vue **Basic settings**.

2. Sélectionnez **Built-In** dans la liste **Schema**. Cliquez sur **Edit schema** afin de définir la même structure des données que dans le schéma du **tHashOutput**.

3. Sélectionnez **tHashOutput_2** dans la liste **Component list**.

4. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.

5. Sélectionnez **Built-In** dans la liste **Schema** et cliquez sur **Sync columns** pour récupérer le schéma du composant précédent. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)**.

Sauvegarder et exécuter le Job

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** pour exécuter le Job, ou cliquez sur le bouton **Run**, dans l'onglet **Run**.

Vous pouvez constater qu'une seule ligne a été écrite alors que deux ont été générées par le **tFixedFlowInput**.

tHashOutput

Propriétés du tHashOutput

Les composants de la famille Technical sont normalement masqués par défaut, dans la **Palette**. Pour savoir comment les afficher dans la **Palette**, consultez le *Guide utilisateur du Studio Talend*.

Famille de composant	Technical	
Fonction	Le composant **tHashOutput** écrit des données dans la mémoire cache, afin de permettre un accès rapide aux données.	
Objectif	Ce composant charge des données dans la mémoire cache afin de permettre un accès rapide aux données et de faciliter les transactions comportant un large volume de données.	
Basic settings	*Schema* et *Edit schema*	Un schéma est une description de ligne, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant (Repository). Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.
		Built-in : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Link with a tHashOutput	Cochez cette case pour connecter votre composant à un **tHashOutput**. Si plusieurs composants **tHashOutput** sont ainsi reliés, les données chargées dans le cache par tous ces composants peuvent être lues par un **tHashInput** relié à l'un d'entre eux.
	Component list	Liste déroulante comprenant les composants **tHashOutput** disponibles.
	Data write model	Liste des modes d'écriture de données disponibles.
	Keys management	Liste des modes de gestion de clés disponibles. • **Keep all** : écrit toutes les données reçues dans la mémoire cache. • **Keep first** : écrit uniquement le premier enregistrement dans la mémoire cache si plusieurs enregistrements sont reçus avec la même valeur de clé.

	Append	Cette case est cochée par défaut. Elle sert à écrire des données dans la mémoire, à la suite, au cas où une itération existe dans le même sous-job. Si la case est décochée, le composant **tHashOutput** va effacer la mémoire avant d'y charger les données. Si la case **Link with a tHashOutput** est cochée, la case **Append** est masquée mais reste activée.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl +Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant écrit des données dans la mémoire cache. Il est étroitement lié au composant **tHashInput**. Ensemble, ces composants permettent un accès rapide aux données et facilitent les transactions comportant un large volume de données.	

Scénarios associés

Pour un scénario associé, consultez :

- *Scénario 1 : Lire des données directement dans la mémoire cache afin d'y accéder rapidement*.
- et *Scénario 2 : Vider la mémoire avant d'y charger les données si une boucle existe dans le même sous-job* du composant **tHashInput**.

Composants XML

Ce chapitre passe en revue les propriétés des principaux composants présents dans la famille **XML** de la **Palette** dans la perspective **Integration** du *Studio Talend*.

La famille **XML** regroupe les composants dédiés aux tâches liées au XML, notamment le parsing, la validation, la création de structures XML, etc.

tAdvancedFileOutputXML

Propriétés du tAdvancedFileOutputXML

Famille de composant	File/Output ou XML	
Fonction	Le composant **tAdvancedFileOutputXML** transmet des données vers un fichier de type XML et propose une interface de gestion des boucles et des Group by si nécessaire.	
Objectif	Le **tAdvancedFileOutputXML** crée un fichier XML contenant les valeurs des données définies dans l'arborescence XML.	
Basic settings	*Property type*	Peut être Built-in ou Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Use Output Stream	Cochez la case pour traiter le flux de données qui vous intéresse. Une fois cochée, le champ **Output Stream** s'affiche et vous pouvez saisir le flux de données souhaité. Le flux de données à traiter doit être ajouté au flux afin que ce composant récupère ces données via la variable représentative correspondante. Cette variable peut être prédéfinie dans votre *Studio Talend* ou fournie par le contexte ou les composants utilisé(s) avec ce composant. Sinon, vous pouvez la définir manuellement et l'utiliser selon votre Job, par exemple à l'aide d'un **tJava** ou d'un **tJavaFlex**. Afin d'éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d'autocomplétion (**Ctrl+Espace**) afin de remplir le champ, si cette variable a été correctement définie. Pour pus d'informations concernant l'utilisation d'un flux, consultez *Scénario 2 : Lire les données d'un fichier distant en mode stream*.
	File name	Chemin d'accès et nom du fichier de sortie, et/ou variable à utiliser.

		Ce champ est indisponible si vous avez coché la case **Use Output Stream**. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Configure XML tree	Ouvre l'interface d'aide à la création du fichier XML. Pour plus d'informations sur l'interface, consultez *Définir un arbre XML*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champ qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Cliquez sur ce bouton pour synchroniser le schéma de sortie avec celui d'entrée. La fonction **Sync** ne s'affiche que si une connexion de type Row est liée au composant de sortie.
	Append the source xml file	Cochez cette option pour ajouter de nouvelles lignes à la fin de votre fichier source XML.
	Generate compact file	Cochez cette case pour générer un fichier ne comprenant aucun espace vide ni aucun séparateur de ligne. L'ensemble des éléments se présente alors sur une ligne unique, ce qui permet de réduire le poids du fichier.
	Include DTD or XSL	Cochez cette case pour ajouter la déclaration DOCTYPE indiquant le nom de l'élément racine et le chemin d'accès et le nom du fichier DTD ou pour ajouter l'instruction de traitement indiquant le type de feuille de style (telle que les XSL) utilisée et son chemin d'accès et nom. **DTD** : saisissez le nom de l'élément racine dans le champ **Root Element** et saisissez le chemin d'accès et le nom du fichier DTD dans le champ **File Name**. **XSL** : saisissez le type de feuille de style dans le champ **Type** et saisissez le chemin d'accès et le nom du fichier dans le champ **File Name**.
Advanced settings	*Split output in several files*	Si le fichier XML de sortie est lourd, vous pouvez le scinder en plusieurs fichiers en définissant un certain nombre de lignes par fichier.
	Trim Data	Cette case est activée lorsque vous utilisez le mode de génération dom4j. Cochez cette case pour supprimer les espaces en début et en fin de valeur d'un élément XML.
	Create directory only if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier XML de sortie s'il n'existe pas déjà.
	Create empty element if needed	Cette case est cochée par défaut. Si le contenu de la colonne **Associated Column** est nul ou si aucune colonne n'est associée au nœud XML, cette option créera une balise ouvrante et une balise fermante aux endroits prévus.
	Create attribute even if its value is NULL	Cochez cette case pour générer l'attribut de la balise XML pour la colonne d'entrée associée dont la valeur est null.
	Create attribute even if it is unmapped	Cochez cette case pour générer l'attribut de la balise pour la colonne d'entrée associée non mappée.
	Create associated XSD file	Si l'un des éléments XML est associé à un espace de nommage, cette option créera le fichier XSD correspondant.

		Pour utiliser cette option, vous devez sélectionner le mode de génération **Dom4J** dans la liste **Generation mode**.
	Add Document type as node	Cochez cette case pour ajouter des colonnes de type **Document** dans le nœud au lieu d'échapper des chaînes de caractère dans le fichier XML de sortie. Cette case apparaît uniquement lorsque le mode de génération est configuré à **Slow and memory-consuming (Dom4j)** dans l'onglet **Advanced settings**.
	Advanced separator (for number)	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Generation mode	Sélectionnez le mode de génération correspondant à votre mémoire disponible. Les modes disponibles sont : • Lent et consommateur de mémoire (**Slow and memory-consuming - Dom4J**). Cette option vous permet d'utiliser Dom4J pour traiter des fichiers XML très complexes. • Rapide et peu consommateur de mémoire (**Fast with low memory consumption**). Lorsque la case **Append the source xml file**, dans la vue **Basic settings** est cochée, ce champ disparaît car le mode de génération dom4j est automatiquement sélectionné.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	Don't generate empty file	Cochez cette case pour annuler la génération du fichier si celui-ci est vide.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Utilisez ce composant pour créer un fichier XML à partir des données transmises par d'autres composants via une connexion de type Row.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Définir un arbre XML

Double-cliquez sur le composant **tAdvancedFileOutputXML** pour ouvrir l'interface dédiée ou cliquez sur le bouton **[...]** du champ **Configure Xml Tree** de l'onglet **Basic settings** dans la vue **Component**.

Le champ **Schema List** de la zone **Linker Source**, à gauche de l'interface, contient toutes les colonnes du flux d'entrée (à condition que le flux d'entrée soit connecté au composant **tAdvancedFileOutputXML**).

A droite de l'interface, dans la zone **Linker Target**, définissez la structure XML que vous souhaitez obtenir en sortie.

Vous pouvez la créer manuellement ou tout simplement importer la structure XML. Puis importez les colonnes du schéma d'entrée dans l'élément de l'arbre XML correspondant.

Importer un arbre XML

Le meilleur moyen de renseigner l'arbre XML est d'importer un fichier XML bien formé.

1. Renommez la balise racine qui s'affiche par défaut dans le panneau **XML tree,** en cliquant sur celle-ci.
2. Dans la colonne **XML Tree**, cliquez-droit sur le champ **root tag** pour afficher le menu contextuel.
3. Dans le menu, sélectionnez **Import XML tree**.
4. Sélectionnez le fichier à importer et cliquez sur **OK**.

- Vous pouvez importer la structure XML d'un fichier au format XML, XSD et DTD.
- Lors de l'import d'une structure XML d'un fichier XSD, vous pouvez choisir un élément comme la racine de votre arbre XML.

XML Tree	Related Column	Node Status
VideoCollection		
Movie		
@id		-
Category		
Year		
Language		
Title		
Crew		
Director		
Cast		

La colonne **XML Tree** est donc automatiquement renseignée avec les éléments. Vous pouvez supprimer et ajouter des éléments ou des sous éléments à l'arbre :

1. Sélectionnez l'élément adéquat dans l'arbre.
2. Cliquez-droit pour faire apparaître le menu contextuel.
3. Sélectionnez **Delete** pour supprimer la sélection de l'arbre ou sélectionnez l'option adéquate parmi les suivantes : **Add sub-element**, **Add attribute**, **Add namespace** pour enrichir l'arbre.

Créer manuellement l'arbre XML

Si vous ne possédez pas de structure XML déjà définie, vous pouvez la créer manuellement.

1. Dans la colonne **XML Tree**, cliquez une fois sur le champ **root tag** pour le renommer.
2. Cliquez-droit sur ce champ pour afficher le menu contextuel.
3. Dans le menu, sélectionnez **Add sub-element** pour créer le premier élément de la structure.

Vous pouvez aussi ajouter un attribut ou un élément enfant à n'importe quel élément de l'arbre ou supprimer n'importe quel élément de l'arbre.

1. Sélectionnez l'élément adéquat dans l'arbre que vous venez de créer.
2. Cliquez-droit à gauche du nom de l'élément pour afficher le menu contextuel.
3. Dans le menu, sélectionnez l'option adéquate parmi les suivantes : **Add sub-element**, **Add attribute**, **Add namespace** ou **Delete**.

Mapping de données XML

Une fois votre arbre XML créé, vous pouvez alimenter chaque élément ou sous-élément XML avec les colonnes du flux d'entrée dans la colonne **Related Column** :

1. Cliquez sur une des entrées de la colonne **Schema List**.

2. Glissez-la dans le sous-élément correspondant à droite.

3. Relâchez-la pour que le mapping soit effectif.

Une flèche bleue apparaît pour illustrer ce mapping. S'il est disponible, utilisez le bouton **Auto-Map** situé en bas à gauche de l'interface, il permet d'effectuer cette opération automatiquement.

Vous pouvez déconnecter n'importe quel mapping de n'importe quel élément de l'arbre XML :

1. Sélectionnez l'élément de l'arbre XML que vous souhaitez déconnecter de sa source respective.

2. Cliquez-droit à gauche de son nom pour afficher le menu contextuel.

3. Sélectionnez **Disconnect linker**.

La flèche bleue disparaît.

Définir le statut du nœud

Définir l'arbre XML et le transfert de données ne suffit pas. Vous devez aussi définir l'élément sur lequel la boucle est effectuée et si nécessaire l'élément à partir duquel le regroupement est effectué.

Elément Boucle

L'élément Boucle permet de définir l'élément objet d'une itération. L'élément Boucle est généralement un générateur de lignes.

Pour définir un élément comme un élément Boucle :

1. Sélectionnez l'élément adéquat dans l'arbre XML.

2. Cliquez-droit à gauche du nom de l'élément pour afficher le menu contextuel.

3. Sélectionnez l'option **Set as Loop Element**.

La colonne **Node Status** affiche le nouveau statut sélectionné.

Il ne peut y avoir qu'un seul élément Boucle à la fois.

Elément Group

L'élément Group est optionnel, il représente un élément constant sur lequel est effectuée une fonction Groupby. L'élément Group ne peut être défini que si un élément Boucle a été préalablement défini.

Lorsque vous utilisez un élément Group, les lignes doivent être triées pour pouvoir être regroupées par le nœud sélectionné.

Pour définir un élément comme élément Group :

1. Sélectionnez l'élément adéquat dans l'arbre XML.
2. Cliquez-droit à gauche du nom de l'élément pour afficher le menu contextuel.
3. Sélectionnez l'option **Set as Group Element**.

La colonne **Node Status** affiche le nouveau statut sélectionné et tout autre statut de regroupement est automatiquement défini, si nécessaire.

Une fois le mapping terminé, cliquez sur **OK** pour valider les paramètres et continuez la configuration de votre Job.

Scénario : Créer un fichier XML à l'aide d'une boucle

Le scénario suivant décrit la création d'un fichier XML à partir d'un fichier plat trié contenant une vidéothèque.

Configurer le fichier source

1. Dans la **Palette** de composants, cliquez sur les composants **tFileInputDelimited** et **tAdvancedFileOutputXML** et déposez-les dans l'espace de modélisation.

2. Sinon, si vous avez déjà créé une métadonnée contenant la description du fichier d'entrée dans le nœud **Metadata** du **Repository**, alors vous pouvez cliquer-déposer directement cette métadonnée dans l'espace de modélisation pour paramétrer automatiquement le flux d'entrée.

3. Cliquez-droit sur le composant d'entrée et reliez-le au composant **tAdvancedFileOutputXML** à l'aide d'une connexion de type **Row Main**.

4. Sélectionnez le composant **tFileInputDelimited** et paramétrez ses propriétés dans la vue **Component** en bas de la fenêtre du Studio.

5. Dans la liste déroulante **Property Type**, sélectionnez l'option **Repository** si la description de votre fichier est stockée dans une métadonnée du Repository. Si vous cliquez-déposez le composant directement à partir de la métadonnée, vous n'aurez pas besoin de modifier ses propriétés.

   Sinon, sélectionnez l'option **Built-In** et renseignez manuellement les champs de l'onglet **Basic settings**.

6. Le fichier d'entrée contient les colonnes suivantes, séparées par des points-virgules : *id*, *name*, *category*, *year*, *language*, *director* et *cast*.

   Dans cette exemple, le champ **Cast** regroupe différentes valeurs et **id** s'incrémente lorsque l'on change de film.

7. Si nécessaire, définissez le schéma du **tFileDelimitedInput** en fonction de la structure du fichier.

8. Une fois que vous avez vérifié que le schéma du fichier d'entrée correspond à vos attentes, cliquez sur **OK** pour valider.

Configurer la sortie XML et le mapping

1. Puis sélectionnez le composant **tAdvancedFileOutputXML** et cliquez sur la vue **Component** pour paramétrer ses propriétés, ainsi que le mapping dans l'onglet **Basic settings**. Notez que vous pouvez directement double-cliquer sur le composant pour ouvrir l'interface de mapping.

2. Dans le champ **File Name**, sélectionnez le fichier qui sera écrit à la fin du Job, s'il existe déjà ou saisissez le chemin d'accès et le nom du fichier qui seront créés automatiquement pour la sortie.

   Par défaut, le schéma (description du fichier) est automatiquement propagé à partir du flux d'entrée. Mais vous pouvez le modifier si nécessaire.

3. Puis cliquez sur le bouton **[...]** ou double-cliquez sur le composant **tAdvancedFileOutputXML** dans l'espace de modélisation pour ouvrir l'éditeur de mapping.

   Les colonnes du schéma du fichier d'entrée sont listées à gauche de l'interface.

4. A droite de l'interface, configurez l'arbre XML afin qu'il reflète la structure du fichier de sortie.

   Vous pouvez soit créer la structure nœud par nœud. Pour plus d'informations sur la création manuelle de l'arbre XML, consultez *Définir un arbre XML*.

   Dans cet exemple, un modèle XML est utilisé pour renseigner automatiquement l'arbre XML.

5. Cliquez-droit sur le champ **root tag** qui apparaît par défaut et sélectionnez l'option **Import XML tree** à la fin des options du menu contextuel.

6. Dans la fenêtre **[Ouvrir]**, sélectionnez le fichier XML à importer et cliquez sur **OK** pour valider l'import.

   Vous pouvez importer une arborescence XML à partir de fichiers au format XML, XSD et DTD.

7. Puis cliquez-déposez chaque champ de la colonne **Schema List** vers son élément correspondant dans l'arbre XML de la colonne **XML tree** tel que décrit dans *Mapping de données XML*.

   Le mapping apparaît sous la forme d'une flèche bleue entre la zone de gauche et celle de droite.

   Enfin, dans la colonne **Node Status**, définissez le statut du nœud sur lequel la boucle est effectuée. Dans cet exemple, l'élément *Cast* correspondant à l'élément Boucle puisque les données de ce champ dans le fichier d'entrée ne sont jamais les mêmes.

   Cliquez-droit sur l'élément **Cast** de l'arbre XML et sélectionnez l'option **Set as loop element**.

8. Pour regrouper les données par film, l'élément **Movie** doit aussi être défini comme élément Group.

   Cliquez-droit sur le nœud parent **Movie** de l'arbre XML et sélectionnez l'option **Set as group element**.

   Le statut sélectionné paraît sur la ligne de l'élément correspondant dans la colonne **Node Status**.

9. Cliquez sur **OK** pour valider la configuration.

10. Appuyez sur **F6** pour exécuter le Job.

VideoCollection2.xml

```
1  <?xml version="1.0" encoding="ISO-8859-15"?>
2
3  <VideoCollection>
4    <Movie id="1">
5      <Category>Comedy</Category>
6      <Year>1982</Year>
7      <Language>English</Language>
8      <Title>Tootsie</Title>
9      <Crew>
10       <Director>Sydney Pollack</Director>
11       <Cast>Dustin Hoffman</Cast>
12       <Cast>Jessica Lange</Cast>
13       <Cast>Sydney Pollack</Cast>
14     </Crew>
15   </Movie>
16   <Movie id="2">
17     <Category>Science Fiction</Category>
18     <Year>1997</Year>
19     <Language>French</Language>
20     <Title>The 5th Element</Title>
21     <Crew>
22       <Director>Eric Besson</Director>
23       <Cast>Bruce Willis</Cast>
```

Le fichier XML de sortie affiche la structure telle qu'elle a été définie.

tDTDValidator

Propriétés du tDTDValidator

Famille de composant	XML	
Fonction	Le composant **tDTDValidator** valide un fichier XML entrant en fonction d'une DTD et envoie les informations de validation à la sortie définie.	
Objectif	Ce composant permet de vérifier les données et la qualité de la structure du fichier traité.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Il contient les informations standard de validation de fichier.
	DTD file	Chemin d'accès au fichier DTD de référence.
	XML file	Chemin d'accès au fichier XML à valider.
	If XML is valid, display If XML is invalid, display	Saisissez un message à afficher dans la console **Run** basé sur le résultat de la comparaison.
	Print to console	Cochez cette case pour afficher le message de validation.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **DIFFERENCE** : résultat de la validation. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **VALID** : résultat de la validation. Cette variable est une variable **Flow** et retourne un booléen. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone mais il est généralement connecté à un composant de sortie pour rassembler les informations de validation.	

Scénario : Valider un fichier XML

Ce scénario décrit un Job permettant de valider plusieurs fichiers d'un répertoire et de transmettre les informations de validation des fichiers non valides dans un fichier délimité.

- Cliquez et déposez les composants suivants de la **Palette** : **tFileList**, **tDTDValidator**, **tMap**, **tFileOutputDelimited**.
- Connectez le **tFileList** et le **tDTDValidator** avec un lien **Iterate** et les composants restants avec des liens **Row Main**.
- Paramétrez les propriétés du composant **tFileList** afin qu'il récupère les fichiers XML du répertoire.

- Dans le champ **Filemask**, saisissez le masque **.xml* entre guillemets doubles.
- Paramétrez le chemin d'accès des fichiers XML à vérifier.
- Décochez la case **Case Sensitive**.
- Dans la vue **Component** du composant **tDTDValidate**, le schéma est en lecture seule puisqu'il contient des informations de validation standard.
- Paramétrez les propriétés du fichier DTD à utiliser comme référence, dans le champ **DTD file**.

- Dans le champ **Xml file**, appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales et sélectionnez la variable permettant de récupérer le chemin d'accès des fichiers : `((String)globalMap.get("tFileList_1_CURRENT_FILE"))`
- Dans les divers messages à afficher dans la console **Run**, utilisez JobName pour afficher le libellé du Job. Rappelez aussi le nom du fichier grâce à la variable : `((String)globalMap.get("tFileList_1_CURRENT_FILE"))`.
- Cochez la case **Print to Console**.
- Dans le composant **tMap**, glissez-déposez les informations de validation du schéma standard que vous souhaitez passer au fichier de sortie.

- Une fois que le schéma de sortie est défini, ajoutez un filtre de condition pour ne sélectionner que les informations lorsque le fichier XML n'est pas valide.
- Saisissez d'abord la valeur souhaitée pour cette variable, l'opérateur en fonction du type de données filtrées puis la variable devant correspondre à cette condition, ici : `0 == $row1[validate]`
- Puis (si cela n'est pas déjà effectué) connectez le **tMap** au **tFileOutputDelimited** via un lien **Row** > **Main**. Nommez-le *errorsOnly*.
- Dans l'onglet **Basic settings** du composant **tFileOutputDelimited**, paramétrez le chemin d'accès de destination du fichier délimité, les séparateurs de champs et l'encodage.
- Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

```
Starting job tDTDValidator_oj at 14:28 14/10/2010.

[statistics] connecting to socket on port 3420
[statistics] connected
[job tDTDValidator_oj]employee_1.xml is Valid
[job tDTDValidator_oj]employee_2.xml is Invalid
[job tDTDValidator_oj]employee_3.xml is Valid
[job tDTDValidator_oj]employee_4.xml is Invalid
[job tDTDValidator_oj]employee_5.xml is Invalid
[statistics] disconnected
Job tDTDValidator_oj ended at 14:28 14/10/2010. [exit
code=0]
```

Dans la console de la vue **Run**, les messages définis apparaissent pour chaque fichier invalide. Au même moment, le fichier de sortie est alimenté avec les informations de validation.

tEDIFACTtoXML

Propriétés du tEDIFACTtoXML

Famille de composant	XML > EDIFACT / Unstructured > EDIFACT	
Fonction	Le composant **tEDIFACTtoXML** lit un message EDIFACT des Nations Unies (Echange de Données Informatisées pour l'Administration, le Commerce et le Transport) et le transforme au format XML selon la version et la famille EDIFACT.	
Objectif	Le **tEDIFACTtoXML** est utilisé pour transformer un fichier de message EDIFACT au format XML afin que les utilisateurs en aient une meilleure lisibilité et que la compatibilité avec les outils de traitement soit meilleure.	
Basic settings	*Schema* et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est fixe et en lecture seule, avec une colonne : *document*.
	EDI filename	Chemin d'accès au fichier de message EDIFACT à transformer.
	EDI version	Sélectionnez dans la liste la version d'EDIFACT du fichier d'entrée.
	Ignore new line	Cochez cette case pour ignorer le retour chariot dans le fichier d'entrée.
	Die on error	Cette case est décochée par défaut et stoppe le Job en cas d'erreur. Cela permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Cochez cette case pour arrêter l'exécution du Job en cas d'erreur.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être lié à un composant de sortie afin de rassembler les résultats de transformation.	

Scénario : Passer d'EDIFACT à XML

Ce scénario décrit un Job simple qui lit un fichier de message EDIFACT Customs Cargo (CUSCAR) et le sauvegarde en tant que fichier XML.

- Déposez un composant **tEDIFACTtoXML** et un **tFileOutputXML** de la **Palette** dans l'espace de modélisation graphique.
- Connectez le composant **tEDIFACTtoXML** au **tFileOutputXML** à l'aide d'un lien **Row** > **Main**.
- Double-cliquez sur le composant **tEDIFACTtoXML** afin d'afficher sa vue **Basic settings**.

- Saisissez dans le champ **EDI filename** le chemin d'accès au fichier du message EDIFACT d'entrée ou parcourez votre système.

  Dans ce scénario, le fichier d'entrée est *99a_cuscar.edi*.
- Dans la liste **EDI version**, sélectionnez la version d'EDIFACT du fichier d'entrée, *D99A* dans cet exemple.
- Cochez la case **Ignore new line** afin d'ignorer les caractères de retour chariot dans le fichier d'entrée durant la transformation.
- Laissez les autres paramètres tels qu'ils sont.
- Double-cliquez sur le composant **tFileOutputXML** afin d'afficher sa vue **Basic settings**.

- Dans le champ **File Name**, saisissez le chemin d'accès au fichier XML de sortie que vous souhaitez générer ou parcourez votre système.

  Dans cet exemple, le fichier XML de sortie est *99a_cuscar.xml*.
- Laissez les autres paramètres tels qu'ils sont.
- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

Le fichier du message d'entrée EDIFACT CUSCAR est transformé au format XML et le fichier XML de sortie est généré comme défini.

99a_cuscar.edi | 99a_cuscar.xml

```
1  UNB+UNOA:2+SENDER+XXXXXLCTA+100421:0437+1918'
2  UNH+163477+CUSCAR:D:99A:UN:SCPRBL'
3  BGM++MOL-EU2-HFA-012W-XXXX8896514-01+VSA'
4  DTM+184:20100429:102'
5  LOC+9+CNYTN'
```

99a_cuscar.edi | 99a_cuscar.xml

```
4  <document><unEdifact>
5      <UNB>
6          <syntaxIdentifier>
7              <id>UNOA</id>
8              <versionNum>2</versionNum>
9          </syntaxIdentifier>
10         <sender>
11             <id>SENDER</id>
12         </sender>
13         <recipient>
14             <id>XXXXXLCTA</id>
15         </recipient>
16         <dateTime>
17             <date>100421</date>
18             <time>0437</time>
19         </dateTime>
20         <controlRef>1918</controlRef>
21     </UNB>
```

tExtractXMLField

Propriétés du tExtractXMLField

Famille de composant	XML	
Fonction	Le composant **tExtractXMLField** lit la structure XML contenue dans les champs d'un fichier ou d'une table et en extrait les données.	
Objectif	Le **tExtractXMLField** lit la structure XML contenue dans les champs d'un fichier ou d'une table, récupère les données contenues dans cette structure et les envoie au composant suivant via un lien Row.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema type et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.
		Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	XML field	Nom du champ XML à traiter. Voir également le *Guide utilisateur du Studio Talend*
	Loop XPath query	Nœud de l'arborescence XML sur lequel est basé la boucle.
	Mapping	**Column** : reflète le schéma tel qu'il est défini dans l'éditeur Edit Schema. **XPath Query** : Saisissez les champs à extraire de la structure Xml d'entrée. **Get nodes** : Cochez cette case pour récupérer le contenu XML de tous les nœuds courants spécifiés dans le champ **Xpath Query**, ou cochez la case correspondant à un nœud spécifique pour ne récupérer que le contenu du nœud sélectionné.
	Limit	Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n'est lue ou traitée.

		Notez que ce champ est indisponible dans la version Map/Reduce du **tExtractXMLField**.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
Advanced settings	*Ignore the namespaces*	Cochez cette case pour ignorer les espaces de noms lors de la lecture et de l'extraction des données XML.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. Notez que ce champ est indisponible dans la version Map/Reduce du **tExtractXMLField**.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être utilisé en composant intermédiaire. Il nécessite donc un composant d'entrée et de sortie. Il permet d'extraire des données d'un champ délimité à l'aide d'un lien **Row > Main**, et permet de créer un flux de rejet avec un lien **Row > Reject**, filtrant les données dont le type ne correspond pas au type défini.	
Utilisation dans des Jobs Map/Reduce	Si vous avez souscrit à l'une des solutions Big Data de **Talend**, ce composant est également disponible en tant que composant Map/Reduce. Dans un Job **Talend** Map/Reduce, il est utilisé comme étape intermédiaire, avec d'autres composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être exécuté directement dans Hadoop. Pour plus d'informations concernant les Jobs **Talend** Map/Reduce, consultez les sections décrivant comment créer, convertir et configurer un Job **Talend** Map/Reduce, dans le *Guide de prise en main de Talend Open Studio for Big Data*. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Scénario 1 : Extraire les données XML d'un champ d'une table de base de données

Ce scénario est composé de trois composants et permet de lire la structure XML contenue dans les champs d'une base de données et d'en extraire les données.

1. Glissez les composants **tMysqlInput** de la famille **Database** > **Mysql**, **tExtractXMLField** de la famille **Xml** et un composant **tFileOutputDelimited** de la famille **File** > **Output** de la **Palette** dans le Job designer.

   Reliez les composants entre eux via des connexions de type **Main** > **Row**.

2. Double-cliquez sur le composant **tMysqlInput** pour paramétrer ses propriétés dans l'onglet **Basic settings** de la vue **Component** :

3. Si votre schéma est déjà stocké sous le nœud **Db Connections** dans le **Repository**, sélectionnez l'option **Repository** dans le champ **Schema** puis choisissez les métadonnées appropriées à partir de la liste.

   Pour plus d'informations concernant le stockage des métadonnées de schéma dans la vue Repository, consultez le *Guide utilisateur du Studio Talend*.

   Si vous n'avez encore défini aucun schéma, sélectionnez l'option **Built-in** dans les listes **Schema** et **Property Type** et saisissez manuellement les informations de connexion et la structure des données dans un schéma. Pour plus d'informations sur les propriétés du composant **tMysqlInput**, consultez *tMysqlInput*.

4. Dans le champ **Table Name**, saisissez le nom de la table dont les champs contiennent les données XML. Dans ce scénario, la table se nomme *customerdetails* et ne contient qu'une seule colonne.

5. Cliquez sur le bouton **Guess Query** pour récupérer la requête correspondant à votre schéma.

6. Dans le Job designer, double-cliquez sur le composant **tExtractXMLField** pour paramétrer ses propriétés dans l'onglet **Basic settings**.

7. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton **[...]** à côté du champ **Edit schema** pour consulter le schéma.

   La colonne **Column** du tableau **Mapping** sera automatiquement renseignée avec la description du fichier que vous avez configuré.

8. Dans la liste **Xml field**, sélectionnez la colonne dont vous souhaitez extraire le contenu XML. La colonne se nomme *CustomerDetails* et est l'unique colonne de la table *customerdetails*.

   Dans le champ **Loop XPath query**, saisissez le nœud de l'arborescence sur lequel effectuer la boucle afin de récupérer les données.

   Dans la colonne **Xpath query**, saisissez entre guillemet le nœud du champ XML contenant les données que vous voulez extraire. Dans ce scénario, l'objectif est d'extraire les noms des clients, saisissez donc *"CustomerName"* qui correspond à l'élément XML contenant cette information.

9. Dans le Job designer, double-cliquez sur le composant **tFileOutputDelimited** pour paramétrer ses propriétés dans l'onglet **Basic settings**.

10. Dans le champ **File Name**, indiquez le répertoire et le nom du fichier de sortie. Pour ce scénario, appelez le fichier *CustomerNames.csv*.

    Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton **[...]** à côté du champ **Edit schema** pour consulter le schéma.

11. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

```
1   Griffith Paving and Sealcoatin
2   Bill's Dive Shop
3   Childress Child Day Care
4   Facelift Kitchen and Bath
5   Terrinni & Son Auto and Truck
6   Kermit the Pet Shop
7   Tub's Furniture Store
8   Toggle & Myerson Ltd
9   Childress Child Day Care
10  Elle Hypnosis and Therapy Cent
11  Lennox Air Pollution Control
```

Le Job retourne les noms des clients contenus sous le nœud `CustomerName` du champ *CustomerDetails* de la table.

Scénario 2 : Extraire les données valides et les données erronées à partir d'un champ XML dans fichier délimité

Ce scénario est composé de quatre composants et permet de lire la structure XML contenue dans les champs d'un fichier délimité, d'en extraire les données et de rejeter les données erronées.

1. A partir de la **Palette**, glissez dans le Job designer les composants **tFileInputDelimited** de la famille **File** > **Input**, **tExtractXMLField** de la famille **Xml**, **tFileOutputDelimited** de la famille **File** > **Output** et **tLogRow** de la famille **Logs & Errors**.

   Connectez les trois premiers composants à l'aide de liens **Row** > **Main**

   Reliez le **tExtractXMLField** au **tLogRow** à l'aide d'un lien **Row** > **Reject**.

2. Double-cliquez sur le composant **tFileInputDelimited** pour paramétrer ses propriétés dans l'onglet **Basic settings**.

3. Sélectionnez l'option **Built-in** à partir de la liste déroulante **Schema** et renseignez manuellement les informations concernant le fichier délimité dans les champs correspondants.

   Cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour afficher la boîte de dialogue permettant de définir la structure des données, comme présenté ci-dessus.

   Cliquez sur le bouton **[+]** pour ajouter autant de lignes que nécessaire à la structure des données. Dans cet exemple, il n'y a qu'une seule colonne dans le schéma : *xmlStr*.

   Cliquez sur **OK** pour valider vos changements et fermer la boîte de dialogue.

   Si vous avez préalablement stocké votre schéma dans le répertoire **Metadata** sous **File delimited**, sélectionnez l'option **Repository** dans la liste déroulante **Schema** puis cliquez sur le bouton **[...]** à côté du champ pour afficher la boîte de dialogue **[Repository Content]** à partir de laquelle vous pouvez sélectionner un schéma pertinent. Cliquez sur **OK** pour fermer la boîte de dialogue et remplir les champs automatiquement avec les métadonnées contenues dans le schéma.

   Pour plus d'informations sur le stockage des schémas de métadonnées dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

4. Dans le champ **File Name**, renseignez le chemin d'accès ou cliquez sur le bouton **[...]** pour parcourir vos dossiers jusqu'au fichier à traiter. Dans cet exemple, il s'agit du fichier *CustomerDetails_Error*.

   Ce fichier csv stocké localement contient une suite de lignes simples au format XML, séparées par un double retour chariot.

   Dans le champ **Row Separator**, renseignez le séparateur de lignes, ici le double retour chariot. Dans le champ **Field Separator**, renseignez le séparateur de champ, aucun ici.

   Dans cet exemple, il n'y a pas d'en-tête (**Header**), de pied de page (**Footer**) ou de nombre limite de lignes à traiter (**Limit**).

5. Dans l'espace graphique, double-cliquez sur le composant **tExtractXMLField** pour paramétrer ses propriétés dans l'onglet **Basic settings**.

6. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton **[...]** à côté du champ **Edit schema** pour consulter le schéma ou le modifier.

   La colonne **Column** du tableau **Mapping** sera automatiquement renseignée avec le schéma défini.

7. Dans la liste **Xml field**, sélectionnez la colonne dont vous souhaitez extraire le contenu XML. Dans cet exemple, la colonne se nomme *xmlStr*.

   Dans le champ **Loop XPath query**, saisissez le nœud de l'arborescence sur lequel effectuer la boucle afin de récupérer les données.

8. Dans le Job designer, double-cliquez sur le composant **tFileOutputDelimited** pour paramétrer ses propriétés dans l'onglet **Basic settings**.

9. Dans le champ **File Name**, indiquez le répertoire et le nom du fichier de sortie. Pour ce scénario, appelez le fichier *CustomerNames_right.csv*.

   Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton **[...]** à côté du champ **Edit schema** pour consulter le schéma ou le modifier.

10. Dans le Job designer, double-cliquez sur le composant **tLogRow** pour paramétrer ses propriétés dans l'onglet **Basic settings** de la vue **Component**.

11. Cliquez sur le bouton **Sync Columns** pour récupérer le schéma du composant précédent. Pour plus d'informations sur le composant **tLogRow**, consultez *tLogRow*.

12. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

```
Starting job A at 14:22 05/11/2009.

[statistics] connecting to socket on port 3845
[statistics] connected
|
<CustomerDetails>
  <CustomerName>Childress Child Day Care</Customername>
  <id>9</id>
</CustomerDetails>||Error on line 3 of document  : The element type
"CustomerName" must be terminated by the matching end-tag "</CustomerName>".
Nested exception: The element type "CustomerName" must be terminated by the
matching end-tag "</CustomerName>". - Line: 8
[statistics] disconnected
Job A ended at 14:22 05/11/2009. [exit code=0]
```

Le composant **tExtractXMLField** lit et extrait dans un fichier délimité en sortie, *CustomerNames_right* dans cet exemple, les informations concernant les clients dont la structure XML est correcte, et affiche les lignes erronées dans la console de log.

tFileInputXML

Propriétés du tFileInputXML

Famille du composant	File/Input	
Fonction	Le composant **tFileInputXML** lit un fichier XML structuré ou un flux de données et extrait les données ligne par ligne. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job Map/Reduce **Talend** afin de générer du code Map/Reduce. Pour plus d'informations, consultez *Le tFileInputXML dans des Jobs Map/Reduce Talend*.	
Objectif	Ce composant lit un fichier XML structuré ou un flux de données ligne par ligne pour le scinder en champs et envoie les champs tel que définis dans le schéma au composant suivant du Job via une connexion **Row**.	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	File name/Stream	**File name** : Chemin d'accès et nom du fichier à traiter. **Stream** : Flux de données à traiter. Les données doivent préalablement être mises en flux afin d'être récupérées par le **tFileInputXML** via la variable représentative correspondante. Cette variable peut être prédéfinie dans votre *Studio Talend* ou fournie par le contexte ou les composants utilisé(s) avec ce composant, par exemple la variable *INPUT_STREAM* du **tFileFetch**. Sinon, vous pouvez la définir manuellement et l'utiliser selon votre Job, par exemple à l'aide d'un **tJava** ou d'un **tJavaFlex**.

		Afin d'éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d'autocomplétion (**Ctrl+Espace**) afin de remplir le champ, si cette variable a été correctement définie. Pour plus d'informations concernant les flux d'entrée, consultez *Scénario 2 : Lire les données d'un fichier distant en mode stream*.
	Loop XPath query	Nœud de l'arborescence sur lequel la boucle est basée.
	Mapping	**Column** : Colonnes à mapper. Elles reflètent le schéma défini par le champ **Schema type**. **XPath Query** : Saisissez les champs à extraire de l'entrée structurée. **Get nodes** : Cochez cette case pour récupérer le contenu XML de tous les nœuds courants spécifiés dans le champ **Xpath Query**, ou cochez la case correspondant à un nœud spécifique pour ne récupérer que le contenu du nœud sélectionné. Ces nœuds sont importants lorsque le flux de sortie de ce composant doit utiliser la structure XML, par exemple pour le type de données **Document**. Pour plus d'informations, consultez le *Guide utilisateur du Studio Talend*. L'option **Get Nodes** fonctionne en mode **DOM4j** et **SAX**, cependant avec le mode **SAX**, les espaces de nommage ne sont pas supportés. Pour plus d'informations concernant les modes **DOM4j** et **SAX**, consultez les propriétés de la liste **Generation mode** de l'onglet **Advanced settings**.
	Limit	Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n'est lue ou traitée. Si -1, toutes les lignes sont lues ou traitées, quelque soit le nombre.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
Advanced settings	*Ignore DTD file*	Cochez cette case afin d'ignorer le fichier DTD indiqué dans le fichier XML traité.
	Advanced separator (for number)	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Ignore the namespaces	Cochez cette case pour ignorer les espaces de noms. **Generate a temporary file** : renseignez le chemin d'accès et le nom du fichier temporaire XML.
	Use Separator for mode Xerces	Cochez cette case pour séparer les valeurs des nœuds fils concaténées. Ce champ ne peut être utilisé que si vous avez sélectionné le mode **Xerces** dans le champ **Generation mode**. Le champ **Field separator** apparaît : définissez le séparateur à utiliser pour séparer les valeurs

	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
	Generation mode	Sélectionnez le mode de génération correspondant à votre mémoire disponible. Les modes disponibles sont : • Lent et consommateur de mémoire (**Slow and memory-consuming - Dom4J**). Cette option vous permet d'utiliser Dom4J pour traiter des fichiers XML très complexes. • Consommateur de mémoire (**Memory-consuming (Xerces)**) • Rapide et peu consommateur de mémoire (**Fast with low memory consumption (SAX)**)
	Validate date	Cochez cette case pour vérifier strictement le format de la date par rapport au schéma d'entrée.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être utilisé en tant que composant d'entrée. Il permet de créer un flux de données XML à l'aide d'un lien **Row** > **Main**, ainsi que de créer un flux de rejet avec un lien **Row** > **Reject** filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d'utilisation de ces deux liens, consultez *Scénario 2 : Extraire les données XML erronées dans un flux de rejet*.	
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).	

Le tFileInputXML dans des Jobs Map/Reduce Talend

*Les informations contenues dans cette section concernent uniquement les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend** et ne sont pas applicables aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job Map/Reduce **Talend**, le **tFileInputXML**, ainsi que le Job complet utilisant ce composant, génère du code Map/Reduce natif. Cette section présente les propriétés spécifiques du **tFileInputXML** lorsqu'il est utilisé dans un Job Map/Reduce. Pour plus d'informations concernant un Job Map/Reduce **Talend**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	MapReduce/Input	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.

		Repository : Réutilisez les propriétés stockées centralement sous le nœud **Hadoop Cluster** de la vue **Repository**. Les champs suivants sont alors pré-remplis à l'aide des données collectées. Pour plus d'informations concernant le nœud **Hadoop Cluster**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.
	Folder/File	Parcourez votre système ou saisissez le chemin d'accès au répertoire de HDFS dans lequel sont stockées les données à utiliser. Si le chemin d'accès défini pointe vers un dossier, ce composant lit tous les fichiers stockés dans le dossier, par exemple */user/talend/in*. Si des sous-dossiers existent, ils sont automatiquement ignorés à moins que le chemin défini soit */user/talend/in/**. Si vous souhaitez spécifier plusieurs fichiers ou dossiers dans ce champ, séparez les chemins à l'aide d'une virgule (,). Si le fichier à lire est un fichier compressé, saisissez son nom ainsi que son extension. Le **tHDFSInput** décompresse le fichier automatiquement à l'exécution. Les formats de compression, ainsi que les extensions correspondantes, sont : • DEFLATE : **.deflate* • gzip : **.gz* • bzip2 : **.bz2* • LZO : **.lzo* Assurez-vous que la connexion à la distribution Hadoop utilisée est correctement configurée dans l'onglet **Hadoop configuration** de la vue **Run**.
	Element to extract	Saisissez l'élément duquel lire le contenu ainsi que les éléments fils des données XML d'entrée. L'élément défini dans ce champ est utilisé au nœud racine d'un XPath spécifié dans le composant. Cet élément vous permet de définir les unités atomiques des données XML à utiliser afin que,

		quelle que soit la taille du document original ou quel que soit l'endroit où l'entrée est divisée, les lignes dans l'élément puissent correctement être distribuées aux tâches du mappeur. Tout contenu en dehors de cet élément est ignoré et les éléments fils de cet élément ne peuvent contenir l'élément lui-même.
	Loop XPath query	Nœud de l'arborescence sur lequel se base la boucle. Sa racine est l'élément défini dans le champ **Element to extract**.
	Mapping	**Column** : colonnes à mapper. Elles reflètent le schéma comme défini dans le champ **Schema type**. **XPath Query** : saisissez les champs à extraire de l'entrée structurée. **Get nodes** : cochez cette case pour récupérer le contenu XML de tous les nœuds courants spécifiés dans la liste **Xpath query**, ou cochez la case à côté des nœuds XML spécifiques afin de récupérer uniquement le contenu des nœuds sélectionnés. Ces nœuds sont importants lorsque le flux d'entrée de ce composant doit utiliser la structure XML, par exemple le type de données **Document**. Pour plus d'informations concernant le type Document, consultez le *Guide utilisateur de Studio Talend*.
	Die on error	Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées,vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.
Advanced settings	*Ignore the namespaces*	Cochez cette case pour ignorer les espaces de nommage.
	Custom encoding	Il est possible de rencontrer des problèmes d'encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste **Encoding**. Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Usage	Compte tenu des caractéristiques du framework MapReduce, la version Map/Reduce du **tFileInputXML** ne supporte pas les parsers XML suivants : les parsers basés DOM, les parsers basés SAX et les parsers basés sur des flux. Dans un Job **Talend** Map/Reduce, ce composant est utilisé comme composant de début et requiert un composant de transformation comme lien de sortie. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Once a Map/Reduce Job is opened in the workspace, **tFileInputXML** as well as the MapReduce family appears in the **Palette** of the Studio.	

	Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.

Scénario 1 : Extraire des adresses XML

Ce scénario est constitué de deux composants : un composant **tFileInputXML** qui extrait des données du fichier XML contenant des adresses et un composant **tLogRow** qui affiche le résultat de cette opération dans la console **Run** via le composant **tLogRow**.

1. Sélectionnez un composant **tFileInputXML** dans le dossier **File** de la **Palette**. Cliquez et déposez également un composant **tLogRow** de la famille **Logs & Errors**.

2. Connectez les deux composants à l'aide d'un lien **Row** > **Main**.

3. Dans l'onglet **Basic settings** du composant **tFileInputXML**, définissez ses propriétés.

4. Le fichier XML contenant les adresses a été préalablement défini dans la zone **Metadata**, par conséquent récupérez ses propriétés en sélectionnant **Repository** dans **Property type**. De cette manière, les propriétés sont automatiquement récupérées et le reste des champs est renseigné (à l'exception du Schéma). Pour plus d'informations concernant l'assistant de création de métadonnées, consultez le *Guide utilisateur du Studio Talend*.

5. De la même manière, sélectionnez le schéma approprié dans la liste de métadonnées du **Repository**. Cliquez sur **Edit schema** si vous souhaitez modifier le schéma chargé.

6. Dans le champ **Filename**, indiquez le fichier structuré d'entrée.
7. Dans le champ **Loop XPath query**, modifiez le nœud de boucle si nécessaire.
8. Dans la table de **Mapping**, renseignez les champs à extraire et à afficher dans la sortie.
9. Si le fichier est volumineux, définissez un nombre de ligne maximum à traiter dans le champ **Limit**.
10. Saisissez l'encodage si nécessaire, puis double-cliquez sur **tLogRow** pour définir le caractère séparateur.
11. Enfin, appuyez sur **F6** ou cliquez sur le bouton **Run** dans la vue **Run** pour exécuter le Job. Dans la console, les champs définis sont extraits du fichier XML structuré et sont affichés.

```
Starting job XMLStreetFinder at 12:42 05/01/2007.
Paris|2eme arrondissement|Rue de la Paix
Paris|8eme arrondissement|Champs Elysees
New York City|Manhattan|Madison avenue
New York City|Brooklyn|Washington heights
Job XMLStreetFinder ended at 12:42 05/01/2007. [exit code
```

Scénario 2 : Extraire les données XML erronées dans un flux de rejet

Ce scénario décrit un Job à trois composants qui permet de lire un fichier XML et :

1. d'une part, de retourner les données XML correctes dans un autre fichier XML,
2. et d'autre part, d'afficher dans la console de log les données erronées dont le type ne correspond pas au type défini.

1. A partir de la **Palette**, glissez les composants **tFileInputXML**, **tFileOutputXML** et **tLogRow** dans l'espace de modélisation.

   Cliquez-droit sur le composant **tFileInputXML**, sélectionnez **Row** > **Main** dans le menu contextuel, et cliquez sur le composant **tFileOutputXML** pour créer le lien.

   Cliquez-droit sur le composant **tFileInputXML**, sélectionnez **Row** > **Reject** dans le menu contextuel, et cliquez sur le composant **tLogRow** pour créer le lien de rejet.

2. Double-cliquez sur le composant **tFileInputXML** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.

3. Dans la liste **Property Type**, sélectionnez **Repository** et cliquez sur le bouton **[...]** pour afficher la boîte de dialogue **[Repository Content]** à partir de laquelle vous pouvez sélectionner la métadonnée relative à votre fichier si vous l'avez stocké sous le nœud **File xml** du répertoire **Metadata** du **Repository**. Les champs suivants seront renseignés automatiquement. Sinon, sélectionnez le mode **Built-in** et renseignez ces champs manuellement.

   Pour plus d'informations sur le stockage des schémas de métadonnées dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

4. Dans le champ **Schema**, sélectionnez **Repository** et cliquez sur le bouton **[...]** pour sélectionner le schéma décrivant la structure de votre fichier si vous l'avez stocké dans le **Repository**. Sinon, cliquez sur le bouton **[...]** à côté du champ **Edit schema** et définissez le schéma manuellement.

   Dans cet exemple, le schéma comprend cinq colonnes : *id*, *CustomerName*, *CustomerAddress*, *idState* et *id2*.

5. Cliquez sur le bouton **[...]** à côté du champ **Filename** et indiquez le chemin d'accès au fichier XML.

6. Dans le champ **Loop XPath query**, saisissez, entre guillemets, le chemin du nœud sur lequel effectuer une boucle.

Dans la table **Mapping**, la colonne **Column** est automatiquement renseignée avec les colonnes définies dans le schéma.

Dans la colonne **XPath query**, saisissez entre guillemets le nœud du fichier XML contenant les données que vous voulez extraire dans la colonne correspondante.

7. Dans le champ **Limit**, saisissez le nombre de lignes à traiter. Dans ce scénario, l'objectif est de traiter les 10 premières lignes.

8. Double-cliquez sur le composant **tFileOutputXML** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.

9. Cliquez sur le bouton **[...]** à côté du champ **File Name** pour sélectionner le répertoire de destination du fichier et saisissez le nom du fichier. Dans ce scénario, appelez le fichier *customer_data.xml*.

   Dans le champ **Row tag**, saisissez entre guillemets le nom que vous souhaitez donner à la balise contenant les données.

   Cliquez sur **Edit schema** et assurez-vous que le schéma est récupéré du schéma d'entrée. Si nécessaire, cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.

10. Double-cliquez sur le composant **tLogRow** pour afficher l'onglet **Basic settings** de la vue **Component** et paramétrer ses propriétés.

    Cliquez sur **Edit schema** et assurez-vous que le schéma est récupéré du schéma d'entrée. Si nécessaire, cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.

    Dans la zone **Mode**, sélectionnez l'option **Vertical**.

11. Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job tFileInputXML at 14:34 06/11/2009.

[statistics] connecting to socket on port 3996
[statistics] connected
.-----------------------------------------------------.
|                   #1. tLogRow_1                     |
+-----------------+-----------------------------------+
| key             | value                             |
+-----------------+-----------------------------------+
| id              | null                              |
| CustomerName    | null                              |
| CustomerAddress | null                              |
| idState         | null                              |
| id2             | null                              |
| errorCode       | null                              |
| errorMessage    | For input string: "ab" - Line: 1  |
+-----------------+-----------------------------------+
.-----------------------------------------------------.
|                   #2. tLogRow_1                     |
+-----------------+-----------------------------------+
| key             | value                             |
+-----------------+-----------------------------------+
| id              | 9                                 |
| CustomerName    | Childress Child Day Care          |
| CustomerAddress | 788 Tennyson Ave.                 |
| idState         | 12                                |
| id2             | null                              |
| errorCode       | null                              |
| errorMessage    | For input string: "cd" - Line: 8  |
+-----------------+-----------------------------------+
[statistics] disconnected
Job tFileInputXML ended at 14:35 06/11/2009. [exit code=0]
```

Le fichier *customer_data.xml* contenant les données correctes est créé et les données erronées sont affichées dans la console de la vue **Run**.

tFileOutputXML

Propriétés de tFileOutputXML

Famille de composant	XML ou File/Output	
Fonction	Le composant **tFileOutputXML** produit en sortie des données sous forme de fichier de type XML. Si vous avez souscrit à l'une des solutions Big Data de **Talend**, vous pouvez utiliser ce composant dans un Job Map/Reduce **Talend** afin de générer du code Map/Reduce. Pour plus d'informations, consultez *Le tFileOutputXML dans des Jobs Map/Reduce Talend*.	
Objectif	Le **tFileOutputXML** écrit dans un fichier XML des données séparées selon un schéma spécifié.	
Basic settings	*File name*	Chemin d'accès et nom du fichier de sortie, et/ou variable à utiliser. Pour plus d'informations concernant l'utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*.
	Incoming record is a document	Cochez cette case si les données du composant précédent sont au format XML. Lorsque cette case est cochée, une liste **Column list** apparaît et vous permet de sélectionner une colonne de type **Document** dans le schéma contenant les données. Le champ **Row tag** disparaît. Lorsque cette case est cochée, dans la vue **Advanced settings**, seules les cases **Create directory if not exists**, **Don't generate empty file**, **Trim data**, **tStatCatcher Statistics** et la liste **Encoding** sont disponibles.
	Row tag	Enveloppe les données et la structure de chaque ligne.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champ qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Cliquez sur ce bouton pour synchroniser le schéma de sortie avec le schéma d'entrée. La fonction Sync ne s'affiche que si une connexion de type Row est liée au composant d'entrée.

Advanced settings	*Split output in several files*	Si le fichier XML en sortie est volumineux, vous pouvez scinder ce fichier en plusieurs, chacun d'entre eux contenant un nombre spécifié de lignes. **Rows in each output file** : Spécifiez le nombre de lignes de chaque fichier de sortie.
	Create directory if not exists	Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.
	Root tags	Spécifiez une ou plusieurs balises racine qui encadrera(ont) la structure et les données du fichier de sortie. La balise racine par défaut est *root*.
	Output format	Définissez le format de sortie. **Column** : Colonnes récupérées du schéma d'entrée. **As attribute** : cochez la case de la colonnes/des colonnes que vous souhaitez utiliser comme attribut(s) de l'élément parent dans la sortie XML. Si la même colonne est sélectionnée dans la table **Output format** comme attribut et dans les paramètres **Use dynamic grouping** comme critère de regroupement dynamique, seul le paramètre **Use dynamic grouping** sera pris en compte pour cette colonne. **Use schema column name** : Par défaut, cette case est cochée pour toutes les colonnes. Ainsi, les libellés des colonnes du schéma d'entrée sont utilisés comme des balises encadrant les données. Si vous souhaitez utiliser une balise différente de celle du schéma d'entrée, décochez cette case et spécifiez une balise entre guillemets, dans le champ **Label**.
	Use dynamic grouping	Cochez cette case si vous souhaitez regrouper dynamiquement les colonnes de sortie. Cliquez sur le bouton **[+]** pour ajouter un ou plusieurs critère(s) de regroupement, dans la table **Group by**. **Column** : Sélectionnez la colonne que vous souhaitez utiliser comme élément d'encadrement pour les lignes de sortie groupées. **Attribute label** : Saisissez entre guillemets un libellé d'attribut pour l'élément d'encadrement du groupe.
	Custom the flush buffer size	Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ **Row number**, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée.
	Advanced separator (for numbers)	Cochez cette option pour modifier les séparateurs utilisés pour les nombres : **Thousands separator** : définissez le séparateur utilisé pour les milliers. **Decimal separator** : définissez le séparateur utilisé pour les décimaux.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
	Don't generate empty file	Cochez cette case pour ne pas générer de fichier vide.
	Trim data	Cochez cette case pour supprimer les espaces en début et fin du texte et fusionner les différents espaces consécutifs en un, dans le texte.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.
Utilisation	Utilisez ce composant pour écrire dans un fichier XML des données récupérées d'autres composants à l'aide d'une connexion de type **Row**.
Log4j	L'activité de ce composant peut être journalisée avec la fonctionnalité *log4j*. Pour plus d'informations sur cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*. Pour plus d'informations sur les niveaux de logs du log4j, consultez la documentation d'Apache : http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).

Le tFileOutputXML dans des Jobs Map/Reduce Talend

*Les informations contenues dans cette section concernent uniquement les utilisateurs ayant souscrit à l'une des solutions Big Data de **Talend** et ne sont pas applicables aux utilisateurs de Talend Open Studio for Big Data.*

Dans un Job Map/Reduce **Talend**, le **tFileOutputXML**, ainsi que le Job Map/Reduce l'utilisant, génèrent nativement du code Map/Reduce. Cette section présente les propriétés spécifiques du **tFileOutputXML** lorsqu'il est utilisé dans cette situation. Pour plus d'informations concernant les Jobs Map/Reduce **Talend**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.

Famille de composant	MapReduce/Output	
Basic settings	*Property type*	Peut être **Built-in** ou **Repository**.
		Built-in : Propriétés utilisées ponctuellement.
		Cliquez sur cette icône pour ouvrir l'assistant de connexion à la base de données et stocker les paramètres de connexion configurés dans la vue **Basic settings**. Pour plus d'informations concernant la configuration et le stockage des paramètres de connexion à la base de données, consultez le *Guide utilisateur du* Studio Talend.
		Repository : permet de réutiliser les propriétés stockées centralement sous le nœud **Hadoop Cluster** de la vue **Repository**. Les champs qui suivent sont pré-renseignés à l'aide des données récupérées. Pour plus d'informations concernant le nœud **Hadoop Cluster**, consultez le *Guide de prise en main de Talend Open Studio for Big Data*.
	Row tag	Spécifiez la balise entourant les données et leur structure par ligne.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champ qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

		• **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job. Voir également le *Guide utilisateur du Studio Talend*.
	Folder	Parcourez votre système ou saisissez le chemin d'accès au répertoire de HDFS dans lequel sont stockées les données à utiliser. Ce chemin doit pointer vers un dossier et non vers un fichier, car les Jobs Map/Reduce **Talend** écrit dans la dossier cible les résultats finaux mais aussi les différents fichiers *part-* générés durant les calculs Map/Reduce. Assurez-vous que la connexion à la distribution Hadoop utilisée est correctement configurée dans l'onglet **Hadoop configuration** de la vue **Run**.
	Action	Sélectionnez une opération pour écrire des données : **Create** : crée un fichier et écrit des données dans ce fichier. **Overwrite** : écrase le fichier existant dans le répertoire spécifié dans le champ **Folder**.
Advanced settings	*Root tags*	Spécifiez une ou plusieurs balise(s) racine pour entourer la structure complète du fichier de sortie et ses données. La balise racine par défaut est *root*.
	Output format	Définissez le format de sortie. **Column** : colonnes récupérées du schéma d'entrée. **As attribute** : cochez la case des colonnes à utiliser comme attributs de l'élément parent dans la sortie XML. Si la même colonne est sélectionnée dans la table **Output format** en tant qu'attribut et dans le paramètre **Use dynamic grouping** comme critère de regroupement dynamique, seul le paramètre de regroupement dynamique est pris en compte pour cette colonne. **Use schema column name**: Par défaut, cette case est cochée pour toutes les colonnes, afin que tous les libellés de colonnes du schéma d'entrée soient utilisés comme balises entourant les données. Si vous souhaitez utiliser une balise différente de celle du schéma d'entrée pour une colonne, décochez cette case pour la colonne en question et spécifiez un libellé entre guillemets dans le champ **Label**.
	Use dynamic grouping	Cochez cette case si vous souhaitez grouper dynamiquement les colonnes de sortie Cliquez sur le bouton **[+]** pour ajouter un ou plusieurs critère(s) de regroupement dans la table **Group by**. **Column** : sélectionnez une colonne que vous souhaitez utiliser comme élément entourant les lignes de sortie groupées.

		Attribute label : saisissez entre guillemets un libellé d'attribut pour l'élément entourant le groupe.
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Dans un Job **Talend** Map/Reduce, ce composant est utilisé en tant que composant de fin et requiert un composant de transformation comme lien d'entrée. Les autres composants utilisés avec lui doivent également être des composants Map/Reduce. Ils génèrent nativement du code Map/Reduce pouvant être directement exécuté dans Hadoop. Once a Map/Reduce Job is opened in the workspace, **tFileOutputXML** as well as the MapReduce family appears in the **Palette** of the Studio. Notez que, dans cette documentation, sauf mention contraire, un scénario présente uniquement des Jobs **Standard**, c'est-à-dire des Jobs **Talend** traditionnels d'intégration de données et non des Jobs Map/Reduce.	
Hadoop Connection	Vous devez utiliser l'onglet **Hadoop Configuration** de la vue **Run** afin de définir la connexion à une distribution Hadoop donnée pour le Job complet. Cette connexion fonctionne uniquement pour le Job dans lequel vous l'avez définie.	

Scénarios associés

Pour des scénarios utilisant le composant **tFileOutputXML**, consultez *Scénario : Transformer un fichier positionnel en fichier XML* et *Scénario 2 : Utiliser un message SOAP depuis un fichier XML pour obtenir le nom d'un pays et le sauvegarder dans un fichier XML*.

tWriteXMLField

Propriétés du tWriteXMLField

Famille de composant	XML	
Fonction	Le composant **tWriteXMLField** lit un fichier XML d'entrée, en extrait la structure et l'insère dans les champs du fichier ou de la table de sortie.	
Objectif	Le **tWriteXMLField** lit un fichier XML d'entrée, extrait la structure du fichier et l'insère dans les champs du fichier ou de la table de sortie.	
Basic settings	*Output Column*	Sélectionnez la colonne du composant de sortie dans laquelle vous souhaitez écrire la structure XML.
	Configure XML Tree	Ouvre l'interface d'aide à la création de la structure XML à écrire dans un champ. Pour plus d'informations sur l'interface, consultez *Définir un arbre XML*.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Depuis la version 5.6, les modes **Built-in** et **Repository** sont disponibles dans n'importe quelle solution de **Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma. • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, sélectionnez **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
		Built-in : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
		Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le *Guide utilisateur du Studio Talend*.
	Sync columns	Cliquez sur ce bouton pour synchroniser le schéma de sortie avec celui d'entrée. La fonction **Sync** ne s'affiche que si une connexion de type Row est liée au composant de sortie.
	Group by	Sélectionnez la colonne à utiliser pour regrouper les données.
Advanced settings	*Remove the XML declaration*	Cochez cette case si vous ne souhaitez pas inclure la déclaration XML.
	Create empty element if needed	Cette case est cochée par défaut. Si le contenu de la colonne **Related Column** de l'éditeur d'arborescence XML est nul, ou si aucune colonne n'est associée au nœud XML, cette option créera une balise ouvrante et une balise fermante aux endroits prévus.
	Expand Empty Element if needed(for dom4j)	Cochez cette case afin de permettre aux éléments null d'être affichés en tant que paires de balises, par exemple <element></element>. Si cette case n'est pas cochées, les éléments null

		sont affichés en tant que balises auto-fermantes, par exemple <element/>. Pour plus d'informations concernant les balises XML, consultez http://www.tizag.com/xmlTutorial/xmltag.php (en anglais). Afin d'utiliser cette option, vous devez sélectionner le mode de génération **Dom4J** . Cette option est disponible lorsque la case **Create empty element if needed** est cochée.
	Create associated XSD file	Si l'un des éléments XML est associé à un espace de nommage, cette option créera le fichier XSD correspondant. Pour utiliser cette option, vous devez sélectionner le mode de génération **Dom4J** dans la liste **Generation mode**.
	Advanced separator (for number)	Cochez cette case pour modifier les séparateurs utilisés par défaut dans les nombres. **Thousands separator** : saisissez entre guillemets le séparateur à utiliser pour les milliers. **Decimal separator** : saisissez entre guillemets le séparateur à utiliser pour les décimales.
	Generation mode	Sélectionnez le mode de génération correspondant à votre mémoire disponible. Les modes disponibles sont : • Lent et consommateur de mémoire (**Slow and memory-consuming - Dom4J**). Cette option vous permet d'utiliser Dom4J pour traiter des fichiers XML très complexes. • Rapide et peu consommateur de mémoire (**Fast with low memory consumption (SAX)**).
	Encoding	Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant doit être utilisé en composant intermédiaire. Il nécessite donc un composant d'entrée et de sortie.	

Scénario : Extraire la structure d'un fichier XML et l'insérer dans les champs d'une base de données

Ce scénario est composé de trois composants et permet de lire un fichier XML, d'en extraire la structure puis de transmettre cette structure dans les champs d'une table.

1. Glissez les composants **tFileInputXml** et **tWriteXMLField** de la famille **Xml** et un composant **tMysqlOutput** de la famille **Database** > **Mysql** de la **Palette** dans l'espace de modélisation graphique.

   Reliez ces composants via des liens de type **Row** > **Main**.

2. Double-cliquez sur le composant **tFileInputXml** pour paramétrer ses propriétés dans l'onglet **Basic settings**.

3. Dans la liste déroulante **Property type**, sélectionnez l'option **Repository** si la description de votre fichier est stockée dans une métadonnée du Repository. Si vous cliquez-déposez le composant directement à partir de la métadonnée, vous n'aurez pas besoin de modifier ses propriétés.

   Pour plus d'informations concernant le stockage des métadonnées dans la vue **Repository** consultez le *Guide utilisateur du Studio Talend*.

4. Sinon, sélectionnez l'option **Built-in** et renseignez manuellement les champs suivants. Pour plus d'informations sur les propriétés du composant **tFileInputXML**, consultez *tFileInputXML*.

   Si vous avez sélectionné l'option **Built-in**, cliquez sur le bouton **[...]** à côté du champ **Edit schema** puis dans la boîte de dialogue, décrivez manuellement la structure de votre fichier.

5. Dans le champ **Look Xpath query**, saisissez sur quel nœud du fichier la boucle doit être effectuée. Pour ce scénario, la boucle est effectuée sur le nœud *customer*. La colonne **Column** du tableau **Mapping** sera automatiquement renseignée avec la description du fichier que vous avez configuré.

   Dans la colonne **Xpath query**, saisissez entre guillemet le nœud du fichier XML contenant les données correspondant à la colonne de gauche.

6. Cliquez sur le composant **tWriteXMLField** dans le Job designer puis cliquez sur la vue **Component** pour l'afficher et paramétrer les propriétés du composant :

7. Cliquez sur le bouton **[...]** à côté du champ **Edit schema** et, dans la boîte de dialogue, ajoutez une ligne en cliquant sur le bouton **[+]**.

8. Dans cette ligne, saisissez le nom de la colonne de votre flux de sortie que vous souhaitez renseigner avec la structure du fichier XML dans la zone de droite, *CustomerDetails* dans ce scénario.

   Dans la colonne **Type** de cette ligne, indiquez qu'elle est de type *String* et dans la colonne **Length**, indiquez qu'elle est de longueur *255*.

   Cliquez sur **OK** pour valider votre schéma de sortie et retourner aux propriétés du composant.

   Dans le champ **Output Column**, sélectionnez la colonne dans laquelle vous souhaitez insérer le contenu XML.

9. Cliquez sur le bouton **[...]** du champ **Configure Xml Tree** pour ouvrir l'interface d'aide à la création de structures XML.

10. Dans la zone **Link Target**, cliquez sur le nœud *rootTag* et renommez-le *CustomerDetails*.

    Dans la zone **Linker source**, sélectionnez *CustomerName* et *CustomerAddress* et glissez-les sur le nœud *CustomerDetails*. Une boîte de dialogue s'ouvre vous demandant quel type d'action vous souhaitez effectuer.

    Sélectionnez **Create as sub-element of target node** afin de créer des sous-éléments au nœud *CustomerDetails*.

    Cliquez-droit sur l'élément *CustomerName* et sélectionnez l'option **Set As Loop Element** dans le menu.

    Cliquez sur **OK** pour valider la structure XML que vous avez définie.

11. Dans le Job designer, double-cliquez sur le composant **tMysqlOutput** pour paramétrer ses propriétés dans l'onglet **Basic settings**.

12. Si votre schéma est déjà stocké sous le nœud **Db Connections** dans le **Repository**, sélectionnez l'option **Repository** dans le champ **Schema** puis choisissez les métadonnées appropriées à partir de la liste.

    Pour plus d'informations concernant le stockage des métadonnées dans la vue **Repository** consultez le *Guide utilisateur du Studio Talend*.

    Si vous n'avez encore défini aucun schéma, sélectionnez l'option **Built-in** et renseignez manuellement les informations de connexion et la structure des données dans un schéma. Pour plus d'informations sur les propriétés du composant **tMysqlOutput**, consultez *tMysqlSCD*.

    Dans le champ **Table**, saisissez le nom de la table dont les champs contiendront les données XML.

Dans le champ **Action on table**, sélectionnez l'opération que vous souhaitez effectuer sur la table. Pour ce scénario, sélectionnez **Create table** pour créer la table.

Dans le champ **Action on data**, sélectionnez l'opération que vous souhaitez effectuer sur les données. Pour ce scénario, laissez l'option **Insert**.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton **[...]** à côté du champ **Edit schema** pour consulter le schéma.

13. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

CustomerDetails
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>talend@apres91</CustomerAddress>...
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>511 Maple Ave. Apt. 1B</CustomerA...
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>662 Lyons Circle</CustomerAddress>...
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>unknown</CustomerAddress>· <Cust...
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>770 Exmoor Rd.</CustomerAddress>·...
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>1860 Parkside Ln.</CustomerAddress...
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>807 Old Trail Rd.</CustomerAddress>...
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>618 Sheriden rd.</CustomerAddress>...
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>788 Tennyson Ave.</CustomerAddre...
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>2032 Northbrook Ct.</CustomerAddre...
<?xml version="1.0" encoding="ISO-8859-15"?>··<CustomerDetails>· <CustomerAddress>4522 N. Greenview Apt. 1B</Custom...

Chaque champ de la colonne **CustomerDetails** est renseignée avec la structure XML du fichier de départ : l'instruction de traitement XML `<?xml version=""1.0"" encoding=""ISO-8859-15""?>`, le premier nœud séparant chaque client `<CustomerDetails>` et les informations sur les clients `<CustomerAddress>` et `<CustomerName>`.

tXMLMap

Le composant **tXMLMap** appartient à deux familles différentes : Processing et XML. Pour plus d'informations, consultez *tXMLMap*.

tXSDValidator

Propriétés du tXSDValidator

Famille de composant	XML	
Fonction	Le composant **tXSDValidator** valide un fichier XML entrant en fonction d'un fichier XSD et envoie les informations de validation à la sortie définie.	
Objectif	Ce composant permet de vérifier les données et la qualité de la structure du fichier traité	
Basic settings	*Mode*	Dans cette liste déroulante, sélectionnez : -**File** pour valider un fichier d'entrée, -**Flow** pour valider un flux d'entrée.
	Schema et *Edit Schema*	Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Il contient les informations standard de validation de fichier.
Mode File uniquement	*XSD file*	Chemin d'accès au fichier XSD de référence. URL HTTP également supportée, par exemple *http://localhost:8080/book.xsd*.
Mode File uniquement	*XML file*	Chemin d'accès au fichier XML à valider.
Mode File uniquement	*If XML is valid, display* *If XML is invalid, display*	Saisissez un message à afficher dans la console **Run** basé sur le résultat de la comparaison.
Mode File uniquement	*Print to console*	Cochez cette case pour afficher le message de validation.
Mode Flow uniquement	*Allocate*	Spécifiez la ou les colonne(s) à valider ainsi que le chemin d'accès au fichier XSD de référence.
Advanced settings	*Encoding*	Saisissez, entre guillemets, le type d'encodage.
	tStatCatcher Statistics	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **DIFFERENCE** : résultat de la validation. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **VALID** : résultat de la validation. Cette variable est une variable **Flow** et retourne un booléen. **XSD_ERROR_MESSAGE** : message d'erreur XSD généré par le composant. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone mais il est généralement connecté à un composant de sortie pour rassembler les informations de validation.	

Scénario : Valider des flux de données par rapport à un fichier XSD

Ce scénario décrit un Job en Java qui valide une colonne XML dans un fichier d'entrée par rapport à un fichier XSD de référence et écrit les informations de log concernant les lignes invalides de la colonne dans un fichier délimité.

Pour un exemple d'utilisation du **tXSDValidator** validant un fichier XML, consultez *Scénario : Valider un fichier XML*.

1. Déposez de la **Palette** dans l'espace de modélisation graphique les composants **tFileInputDelimited**, **tXSDValidator** ainsi que deux **tFileOutputDelimited**.

2. Double-cliquez sur le composant **tFileInputDelimited** afin d'ouvrir sa vue **Component** et définir ses propriétés.

3. Dans la liste **Property Type**, sélectionnez **Built-In**.

   Dans le champ **File name/Stream**, cliquez sur le bouton **[...]** afin de parcourir votre répertoire jusqu'au fichier d'entrée.

   Dans la liste **Schema**, sélectionnez **Built-In**.

   Cliquez sur **Edit Schema** et modifiez le schéma pour qu'il corresponde à celui du fichier d'entrée. Dans ce scénario, le fichier d'entrée contient deux colonnes : *ID* et *ShipmentInfo*. La colonne *ShipmentInfo* est une colonne XML nécessitant d'être validée.

4. Dans l'espace de modélisation graphique, reliez le composant **tFileInputDelimited** au **tXSDValidator** à l'aide d'un lien **Row** > **Main**.

5. Double-cliquez sur le composant **tXSDValidator** et configurez ses propriétés.

6. Dans la liste **Mode**, sélectionnez **Flow Mode**.

   Dans la liste **Schema**, sélectionnez **Built-In**. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

   Pour voir ou modifier le schéma, cliquez sur le bouton **[...]**.Cliquez sur le bouton **[+]** sous le tableau **Allocate** afin d'ajouter une ligne. Le nom de la première colonne du fichier d'entrée apparaît automatiquement dans le champ **Input Column**. Cliquez dans le champ et sélectionnez la colonne que vous souhaitez valider.

   Dans le champ **XSD File**, saisissez le chemin d'accès à votre fichier XSD de référence.

7. Dans l'espace de modélisation, reliez le composant **tXSDValidator** au **tFileOutputDelimited** à l'aide d'un lien **Row** > **Main** afin d'écrire les informations concernant les lignes XML valides.

8. Reliez le **tXSDValidator** au second **tFileOutputDelimited** à l'aide d'un lien **Row** > **Rejects** afin d'écrire les informations concernant les lignes XML invalides.

9. Double-cliquez sur les composants **tFileOutputDelimited** l'un après l'autre et configurez leurs propriétés.

   Dans le champ **File Name**, parcourez votre répertoire jusqu'au fichier que vous souhaitez utiliser.

10. Dans la liste **Schema**, sélectionnez **Built-In** puis cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.

11. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

Les fichiers de sortie contiennent les informations concernant les lignes XML valides et invalides de la colonne spécifiée.

tXSLT

Propriétés du tXSLT

Famille de composant	XML	
Fonction	Le composant **tXSLT** tient compte d'une feuille de style XSL pour transformer un fichier source XML et l'envoyer vers la sortie définie.	
Objectif	Le **tXSLT** permet de transformer une structure de données en une autre structure.	
Basic settings	*XML file*	Chemin d'accès au fichier XML à transformer.
	XSL file	Chemin d'accès au fichier XSL de référence.
	Output file	Chemin d'accès du fichier de sortie. Si le fichier n'existe pas, il sera créé. Le fichier de sortie peut être n'importe quel fichier structuré ou non tels que html, xml, txt ou même pdf ou edifact en fonction de votre fichier xsl.
	Parameters	Cliquez sur le bouton **[+]** pour ajouter de nouvelles lignes à la liste **Parameters** et définir les paramètres de transformation du fichier XSLT. Pour chaque ligne, renseignez la clé dans la colonne *name* et sa valeur associée dans la colonne *value*.
Advanced settings	*tStatCatcher Statistics*	Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables	**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **OUTPUT_FILEPATH** : chemin d'accès au fichier de sortie. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **OUTPUT_FILENAME** : nom du fichier de sortie. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.	
Utilisation	Ce composant peut être utilisé en standalone.	
Limitation	Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Pour plus d'informations, consultez la page https://help.talend.com/display/KB/How+to+install+external+modules+in+the+Talend+products (en anglais) ou la section décrivant comment configurer le studio, dans le *Guide d'installation et de migration Talend*.	

Scénario : Transformer un fichier XML en HTML à l'aide d'une feuille de style XSL

Ce scénario décrit un Job à deux composants appliquant une feuille de style XSL à un fichier XML et génère un fichier HTML en sortie. Il permet également de définir un paramètre de transformation pour la feuille de style XSL afin de changer la couleur de l'arrière plan de l'en-tête dans le document HTML créé en sortie.

1. A partir de la **Palette**, cliquez et déposez un composant **tXSLT** et un **tMsgBox** dans l'espace graphique.

2. Double-cliquez sur le **tXSLT** pour ouvrir la vue **Basic settings** et paramétrer le composant.

3. Dans le champ **XML file**, renseignez le chemin d'accès ou parcourez vos dossiers jusqu'au fichier XML à transformer. Dans cet exemple, il s'agit d'une liste de titres MP3 et leurs informations associées (nom de l'artiste, maison de disque...).

```
1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <catalog>
3   <cd>
4      <title>Empire Burlesque</title>
5      <artist>Bob Dylan</artist>
6      <country>USA</country>
7      <company>Columbia</company>
8      <price>10.90</price>
9      <year>1985</year>
10    </cd>
11    .
12    .
13
14 </catalog>
```

4. Dans le champ **XSL file** de la vue **Basic settings**, renseignez le chemin d'accès ou parcourez vos dossiers jusqu'au fichier XSL approprié.

5. Dans le champ **Output file**, renseignez le chemin d'accès ou parcourez vos dossiers jusqu'au fichier HTML.

6. Dans cet exemple, l'objectif est de convertir les données XML en fichier HTML comprenant un tableau constitué d'un en-tête suivi de la liste du nom des artistes à côté du nom des chansons.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:param name="bgcolor" />

<xsl:template match="/">
  <html>
  <body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr bgcolor="{$bgcolor}">
      <th>Title</th>
      <th>Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
    <tr>
      <td><xsl:value-of select="title"/></td>
      <td><xsl:value-of select="artist"/></td>
    </tr>
    </xsl:for-each>
  </table>
  </body>
  </html>
</xsl:template>

</xsl:stylesheet>
```

7. Dans la zone **Parameters** de la vue **Basic settings**, cliquez sur le bouton **[+]** pour ajouter une ligne où définir le nom et la valeur du paramètre de transformation du fichier XSL. Dans cet exemple, le nom du paramètre de transformation utilisé est *bgcolor* et sa valeur est *green*.

8. Double-cliquez sur le **tMsgBox** pour afficher la vue **Basic settings** et paramétrer le composant.

9. Enregistrez votre Job puis appuyez sur **F6** pour l'exécuter. Une boîte de dialogue s'affiche et vient confirmer que le fichier HTML est bien créé en sortie et qu'il est stocké à l'endroit préalablement défini.

10. Cliquez sur **OK** pour fermer la boîte de dialogue.

Vous pouvez désormais ouvrir le fichier HTML créé pour vérifier la transformation des données XML et la couleur d'arrière-plan de l'en-tête du tableau.

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore