

## Didacticiel – SSDT (SQL Server Data Tools) – Intégration

Cours :	INF 735 – Entrepôt et forage de données
Trimestre :	Automne 2018
Enseignant :	Robert J. Laurin (Chargé de cours) et Michel Hébert (Chargé de laboratoires) <i>(adapté du matériel préparé par Laura Francheri)</i>

Ce didacticiel vous aidera à acquérir les connaissances de base pour utiliser un outil ETC (Extraction, Transformation et Chargement), en créant des packages simples avec **SQL Server Data Tool - Integration Services**.

### Contenu

Didacticiel – SSDT (SQL Server Data Tools) – Intégration .....	1
Prérequis .....	3
Qu'est-ce que SSDT ? .....	3
Composants principaux.....	3
Dans ce didacticiel :.....	4
Pour plus d'information : .....	4
Didacticiel.....	4
Création du package de base .....	5
Création du projet d'intégration.....	5
Création d'un gestionnaire de connexions de fichiers plats.....	7
Création d'un gestionnaire de connexions OLE DB .....	9
Ajout d'une tâche de flux de données .....	10
Ajout et configuration de la source de fichier plat .....	11
Ajout et configuration des transformations de recherche (« lookups ») .....	12
Ajout et configuration de la destination OLE DB .....	15
Exécuter / déboguer le package .....	17
Copie d'un package SSIS .....	19
Redirection de flux d'erreurs .....	19
Création d'un fichier endommagé : .....	20

Copie du package : .....	20
Changement du fichier source : .....	20
Test du package .....	20
Ajout de redirection de flux d'erreurs .....	21
Ajout d'une destination de fichier plat .....	23
Test du package .....	24
Autres didacticiels intéressants : .....	25

## Prérequis

Pour ce didacticiel vous aurez besoin de :

- SQL Server 2012 ou plus récent.
- Microsoft Visual Studio 2015
- Extension SQL Server Data Tools (SSDT)
- Base de données d'exemple « **AdventureWorksDW2012** »
- Données d'exemple fournies au laboratoire

## Qu'est-ce que SSDT ?

- SSDT est l'abréviation de **SQL Server Data Tools**
- Un complément à Visual Studio qui permet de créer des solutions de transformation et intégration de données, d'analyse (cubes) d'entrepôt de données et des rapports.
- Peut se connecter à des bases de données relationnelles, directement ou via ODBC, des fichiers de texte, XML, classeurs Excel, Analysis Services...

Ce laboratoire porte sur les projets d'intégration des données. Le prochain laboratoire portera sur les services d'analyses des entrepôts, soit la création des cubes.

## Composants principaux

Le **package** → unité de travail qui est récupérée, exécutée et enregistrée. C'est l'objet le plus important dans SSIS.

Les **éléments du flux de contrôle** (**tâches** et **conteneurs**) → utilisés pour créer le flux de contrôle dans un package. Un élément du flux de contrôle peut préparer ou copier des données, interagir avec d'autres processus, implémenter un flux de travail répétitif (bouclage), etc. Les **contraintes de précedence** mettent en séquence chaque élément, en permettant de former un flux de contrôle ordonné et en spécifiant les **conditions d'exécution** de tâches ou de conteneurs.

Les **composants du flux de données** (**sources**, **transformations** et **destinations**) → sont utilisés pour créer les flux de données dans un package, qui extraient, transforment et chargent des données. Les chemins d'accès mettent en séquence les composants de manière à former un flux de données ordonné.

Les **gestionnaires de connexions** → se connectent à différents types de sources de données pour extraire et charger des données.

Les **variables** → peuvent être utilisées dans des expressions afin de mettre à jour les valeurs de colonnes et les expressions de propriétés, de contrôler l'exécution de flux de contrôle répétitifs et de définir les conditions appliquées par les contraintes de précedence, tout cela de manière dynamique.

Les **gestionnaires d'événements** → s'exécutent en réponse aux événements d'exécution que les packages, les tâches et les conteneurs déclenchent.

Les **modules fournisseurs d'informations** → prennent en charge l'écriture dans un journal des informations sur l'exécution du package et/ou de chaque tâche ou conteneur, comme l'heure de début et de fin du package, erreurs, etc.

## Dans ce didacticiel :

- On créera des packages simples, on copiera des packages pour les modifier, etc.
- On extraira des données d'un fichier de texte pour les transformer et ensuite les insérer dans une table d'une base de données relationnelle

## Pour plus d'information :

Ce didacticiel est une version modifiée du didacticiel « Création d'un package ETL simple » de Microsoft Technet. La version complète se trouve dans <http://technet.microsoft.com/fr-fr/library/ms169917.aspx>

Un vidéo qui montre le développement d'un package simple : [http://msdn.microsoft.com/fr-fr/library/cc952921\(v=SQL.100\).aspx](http://msdn.microsoft.com/fr-fr/library/cc952921(v=SQL.100).aspx)

## Didacticiel

Pour commencer, on va insérer des données contenues dans le fichier plat **SampleCurrencyData.txt** dans l'entrepôt de données AdventureWorksDW2012. Ensuite, on ajoutera des comportements plus avancés tel que la redirection des enregistrements qui ont des erreurs, etc.

Les **données source** contiennent les quatre colonnes suivantes :

- Taux moyen de la devise
- Devise
- Date
- Taux de clôture.

Exemple :

```
1          USD    1/24/05 0:00    1.001001001
1          USD    1/25/05 0:00    1
1.00040016 USD    1/26/05 0:00    1.001101211
1.00040016 USD    1/27/05 0:00    1.00030009
1.00040016 USD    1/28/05 0:00    1.00070049
1.00040016 USD    1/29/05 0:00    0.99980004
1.00040016 USD    1/30/05 0:00    1.001001001
1.00040016 USD    1/31/05 0:00    0.99990001
```

La table de destination, **FactCurrencyRate** (une table de faits) contient cinq colonnes et des relations avec deux tables de dimension comme illustré ci-après :

Nom de la colonne	Type de données	Table de recherche	Colonne de recherche
CurrencyKey	int (FK)	DimCurrency	CurrencyKey (PK)
DateKey	Int (FK)	DimDate	DateKey (PK)
AverageRate	float	n/a	n/a
EndOfDayRate	float	n/a	n/a
Date	Datetime	n/a	n/a

Malheureusement, cette table n'a pas de colonne de date d'insertion, ce qui rend difficile d'analyser quelles lignes ont été insérées quand... vous pouvez ajouter une colonne de date d'insertion comme suit :

```
ALTER TABLE dbo.FactCurrencyRate ADD
    InsertDate datetime NULL
```

```
ALTER TABLE dbo.FactCurrencyRate ADD CONSTRAINT
    DF_FactCurrencyRate_InsertDate DEFAULT getdate() FOR InsertDate
```

Voici une petite vue à créer pour mieux comprendre les données :

```
CREATE VIEW vFactCurrencyRate AS
SELECT f.CurrencyKey, CurrencyAlternateKey, CurrencyName, FullDateAlternateKey,
    f. DateKey, AverageRate, EndOfDayRate, InsertDate
FROM FactCurrencyRate f
    JOIN DimCurrency c ON (c.CurrencyKey = f.CurrencyKey)
    JOIN DimDate t ON (t. DateKey = f.DateKey)
```

Le mappage de données se fera de la façon suivante :

Colonne de fichier plat	Nom de la table	Nom de la colonne	Type de données
0	FactCurrencyRate	AverageRate	Float
1	DimCurrency	CurrencyAlternateKey	nchar (3)
2	DimDate	FullDateAlternateKey	Date
3	FactCurrencyRate	EndOfDayRate	Float

Puisqu'il s'agit d'un exercice, on doit vérifier que la base de données ne contient pas déjà les données que le va charger dans la table FactCurrencyRate. Nous chargerons les taux de conversion pour les devises USD et VEB. Une recherche dans la table DimCurrency nous indique que les clefs de ces devises sont respectivement 100 et 14. Exécutons donc la commande suivante pour s'assurer que les données correspondantes sont absentes :

```
Delete FactCurrencyRate
where CurrencyKey in (14,100)
```

## Création du package de base

### Création du projet d'intégration

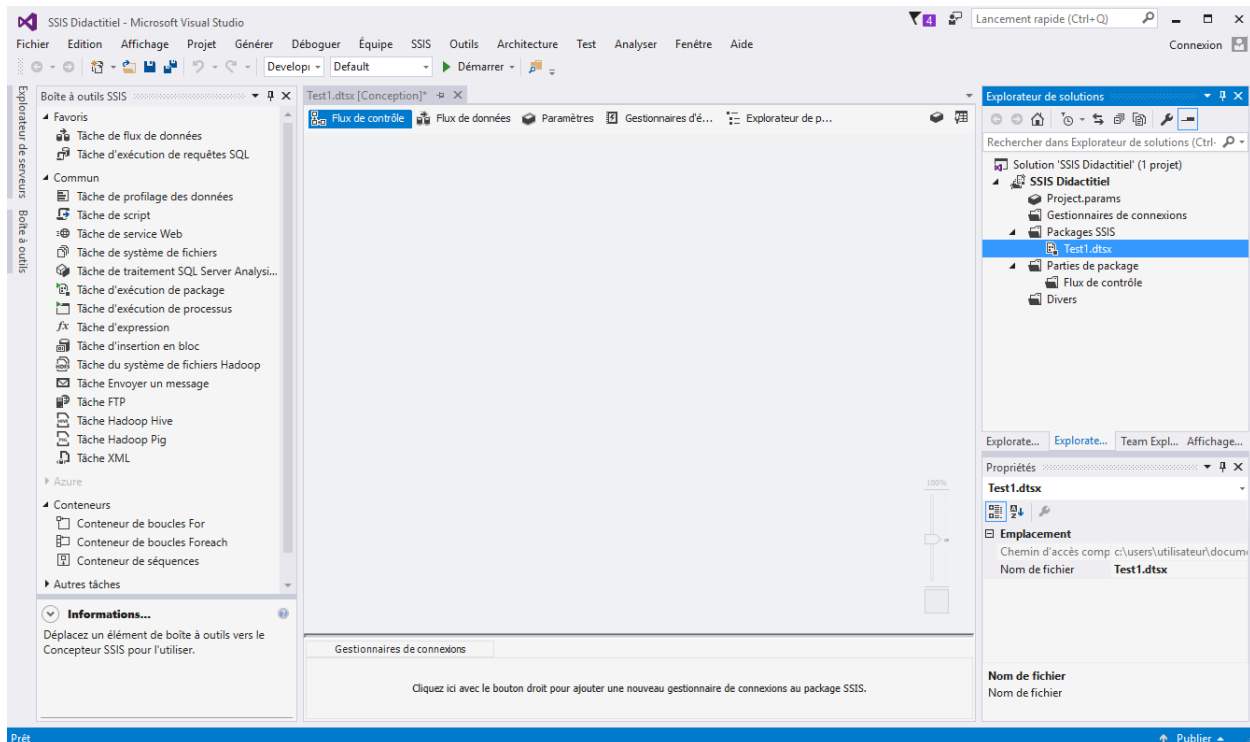
- 1) Ouvrez **MS Visual Studio 2015**.
- 2) Fichier → Nouveau Projet
- 3) Sélectionnez **Projet Integration Services** dans le volet Modèles / Business Intelligence.
- 4) Remplacez le nom par défaut par « SSIS Didacticiel ».
- 5) Acceptez l'emplacement par défaut ou cliquez sur **Parcourir** pour rechercher et accéder au dossier que vous souhaitez utiliser.
- 6) Cliquez sur OK.

Dans l'Explorateur de Solutions, remarquez qu'un package vide, appelé **Package.dtsx**, est créé et ajouté à votre projet.

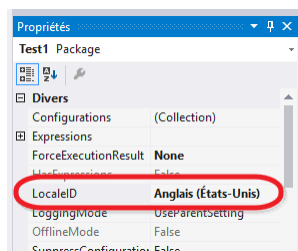
- 7) Bouton droit sur Package.dtsx → Renommer → On appellera le package par défaut « Test1.dtsx ».
- 8) Lorsque le système vous invite à renommer l'objet de package, cliquez sur Oui.

L'écran montre maintenant plusieurs onglets pour le package Test1.dtsx :

- Flux de contrôle
- Flux de données
- Paramètres
- Gestionnaire d'événements
- Explorateur de packages
- Gestionnaires de connexion (dans le bas de l'onglet Flux de contrôle)



- 9) Modifiez les paramètres de configuration régionaux :
  - a. Activez l'onglet Flux de contrôle
  - b. Dans la fenêtre Propriétés, cherchez **LocaleID** et attribuez « Anglais (États-Unis) » pour garantir la compatibilité avec les fichiers d'exemple et la base de données AdventureWorksDW2012.



- Dans la zone de **Flux de contrôle**, on insère les tâches principales du package et on établit leur séquence. On peut aussi rediriger le flux en cas d'erreur, etc.

- Dans la zone de **Flux de données**, on configure les détails des tâches de flux de données.
- Dans la zone de **Gestionnaires de connexions**, on crée des composants qui permettront au package de se connecter à différents sources et destins – soit des fichiers plats, des tables d’une base de données relationnelle, etc.

### Création d’un gestionnaire de connexions de fichiers plats

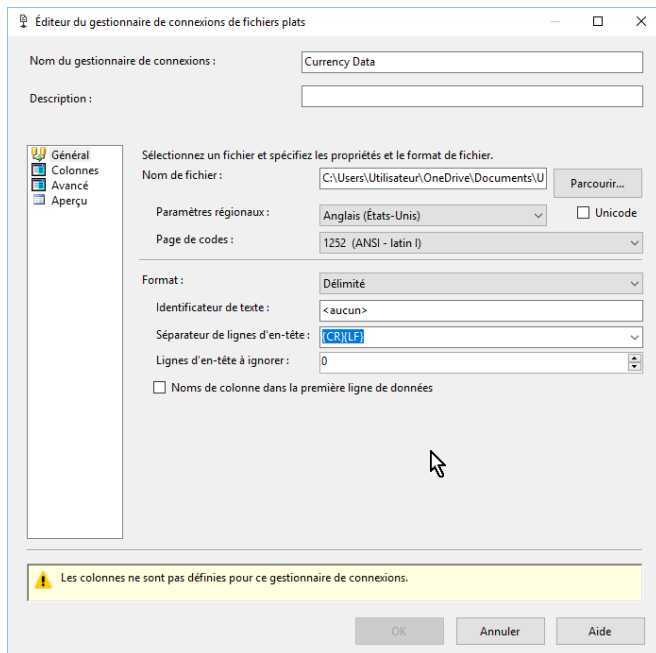
Maintenant, on ajoutera un **Gestionnaire de connexions de fichiers plats** pour que notre package puisse se connecter et extraire des données d’une source de données de type fichier plat. Ce type de gestionnaire de connexion nous permet de spécifier :

- Nom et emplacement du fichier
- Paramètres régionaux et page de codes
- Format du fichier :
  - Séparateur de colonnes,
  - Nom des colonnes
  - Mappage des types de données des colonnes du fichier aux types de données SSIS

**Important** : on doit créer un gestionnaire de connexions de fichiers plats **pour chaque format** de fichier utilisé (mais pas pour chaque fichier individuel !)

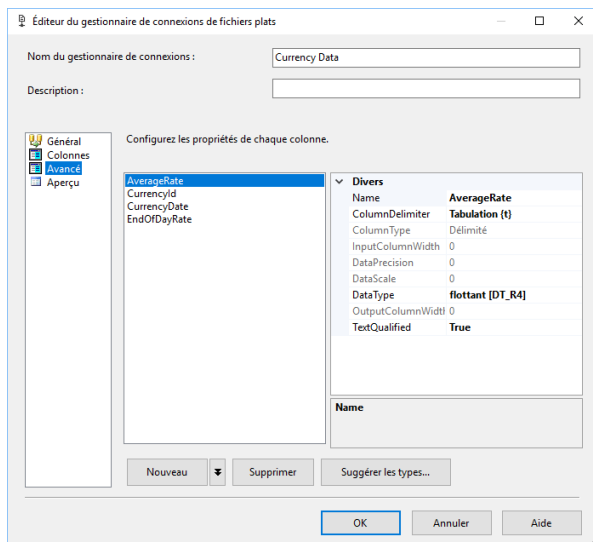
Dans ce cas, on créera un gestionnaire de connexion pour utiliser le fichier **SampleCurrencyData.txt** comme source de données.

- 1) Cliquez avec le bouton droit dans la zone **Gestionnaires de connexions** → Nouvelle connexion de fichier plat.
- 2) Nom du gestionnaire de connexions : entrez Currency Data (par exemple)
- 3) Cliquez sur **Parcourir** pour aller chercher le fichier.
- 4) Ouvrez le fichier **SampleCurrencyData.txt** selon l’endroit où il se trouve.
- 5) **Paramètres régionaux** = Anglais (États-Unis)
- 6) **Page de codes** = 1252 (ANSI – Latin I)
- 7) Enlever le crochet pour **Noms de colonne dans la première ligne de données**



### *Pour définir les colonnes et mappage de données :*

- 8) Dans la liste à gauche, cliquez sur « **Avancé** » :
- 9) Dans le volet des propriétés, à droite, changez les noms des colonnes (propriété **Nom**) par :
  - Column 0 → AverageRate.
  - Column 1 → CurrencyID.
  - Column 2 → CurrencyDate.
  - Column 3 → EndOfDayRate



### *Mappage des types de données*

Remarquez que par défaut, le type de toutes ces colonnes est **chaîne [DT\_STR]** avec une **OutputColumnWidth** de 50.



Vous pouvez éditer la propriété **DataType** et changer les types de données manuellement... ou vous pouvez demander au système de vous aider à choisir les types.

10) Cliquez sur le bouton **Suggérer les types**

11) Pour l'instant, n'apportez aucune modification dans la boîte de dialogue, et cliquez sur OK.

Integration Services suggère automatiquement les types de données adéquats en se basant sur les 200 premières lignes de données. Vous pouvez aussi modifier les options de suggestion et augmenter ou réduire l'échantillon de données, spécifier le type de données par défaut pour les entiers ou les données booléennes ou bien ajouter des espaces pour séparer les colonnes de type chaîne.

Dans ce cas, Integration Services suggère les types de données montrées dans la deuxième colonne de la table ci-dessous.

Toutefois, rappelons que notre objectif est de traiter ces données et de les insérer dans la table **FactCurrencyRate**, c'est-à-dire que ce serait préférable de définir les types de données par rapport aux colonnes de la destination.

La table suivante montre les types suggérés par SSIS dans la 2<sup>e</sup> colonne, et les types de la destination dans la dernière colonne :

Colonne de fichier plat	Type suggéré	Colonne de destination	Type de destination
AverageRate	flottant [DT_R4]	FactCurrencyRate.AverageRate	Float
CurrencyID	chaîne [DT_STR]	DimCurrency.CurrencyAlternateKey	nchar(3)
CurrencyDate	date [DT_DATE]	DimDate.FullDateAlternateKey	dateTime
EndOfDayRate	flottant [DT_R4]	FactCurrencyRate.EndOfDayRate	Float

- (Si les colonnes « flottant » ne sont pas détectées, changez-les à la main ! La suggestion de types n'est pas très intelligente lorsque l'ordinateur n'est pas configuré en anglais...)

Dans ce cas, les colonnes **CurrencyID** et **CurrencyDate** ne sont pas compatibles avec le type de destination.

12) Encore dans le volet Avancé, changez le type de **CurrencyID** pour « **chaîne Unicode [DT\_WSTR]** »

13) Changez le type de **CurrencyDate** pour « **date base de données [DT\_DBDATE]** ».

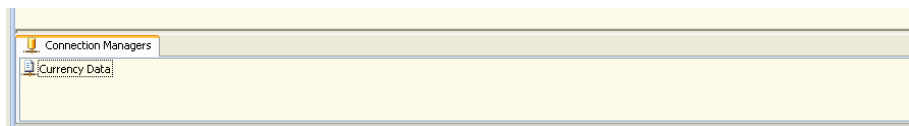
14) Cliquez OK pour finaliser.

### Création d'un gestionnaire de connexions OLE DB

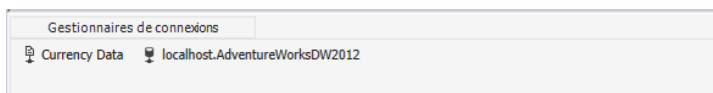
Nous venons de créer le gestionnaire de connexion pour la connexion à la source de données (fichier plat). Maintenant on créera un gestionnaire de connexion pour la destination ; dans notre cas, une base de données relationnelle, compatible OLE DB : « **AdventureWorksDW2012** ».

**Important** : à la différence d'un gestionnaire de connexions pour fichier plat, dont on doit créer un gestionnaire pour chaque format de fichier différent (i.e. pour chaque table qu'on veut charger), on a besoin juste d'un gestionnaire de connexions OLE DB pour accéder à toutes les tables d'une base de données compatible OLE DB, soit en lecture ou en écriture.

- 15) Cliquez avec le bouton droit dans la zone **Gestionnaires de connexions** (en bas) et choisissez **Nouvelle connexion OLE DB**.



- 16) Cliquez sur le bouton **Nouveau**.  
17) Nom du serveur → **localhost** (instance par défaut de SQL Server sur l'ordinateur local – autre nom si vous avez un serveur distant)  
18) Vérifiez si l'option **Utiliser l'authentification Windows** est sélectionnée.  
19) Dans la zone **Sélectionner ou entrer un nom de base de données**, tapez ou sélectionnez **AdventureWorksDW2012**.  
20) Cliquez sur **Tester la connexion**.  
21) Une fois que tout fonctionne correctement, cliquez sur **OK**.  
22) **localhost.AdventureWorksDW2012** apparaît dans la liste de connexions de données.  
23) Assurez-vous de sélectionner cette connexion.  
24) Cliquez sur **OK**.

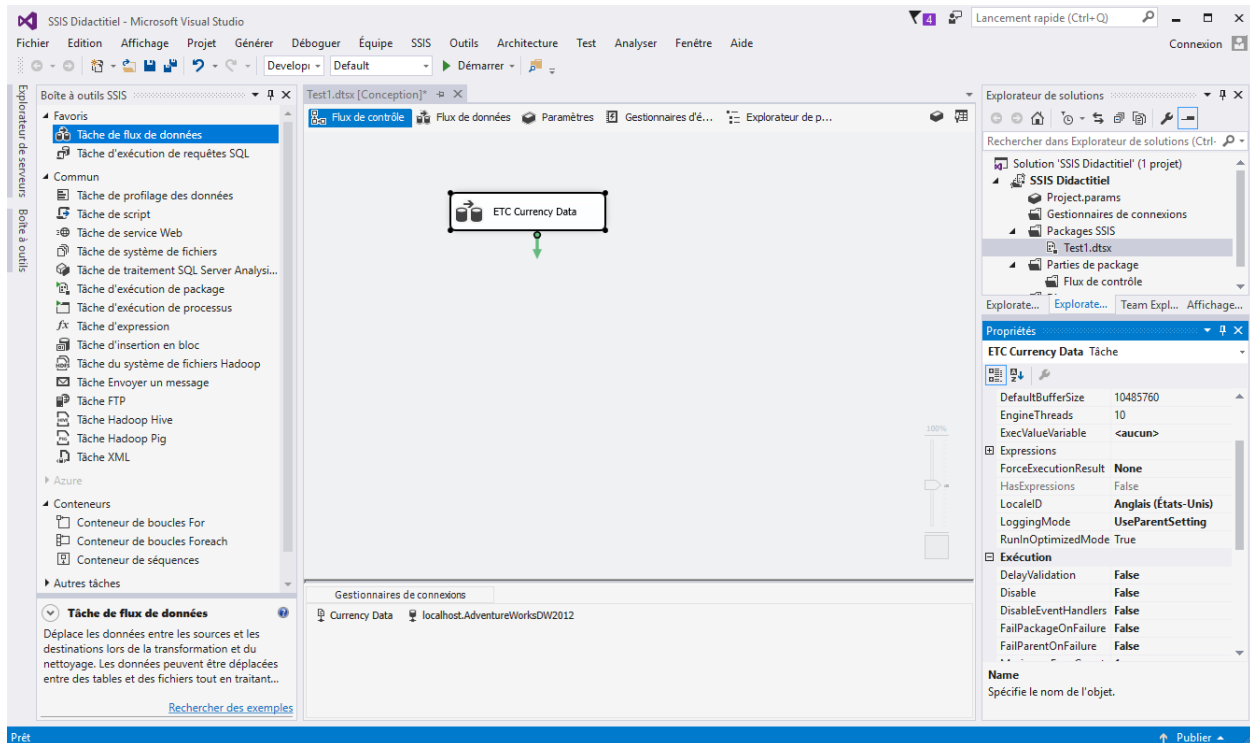


### Ajout d'une tâche de flux de données

Maintenant, on doit extraire les données du fichier source, les transformer et les insérer dans la destination (base de données relationnelle). Pour ce faire, on utilise une **tâche de flux de données**.

La tâche de flux de données est l'endroit où s'effectue la majorité du travail d'un processus d'extraction, de transformation et de chargement (ETC). Elle permet d'encapsuler le moteur de flux de données qui déplace les données entre les sources et les destinations et permet d'effectuer plusieurs transformations, aux fins de nettoyer et modifier les données lors de leur déplacement.

- 25) Cliquez sur l'onglet **Flux de contrôle**.  
26) Dans la Boîte à outils, développez **Favoris**, puis faites glisser une tâche de flux de données dans la zone de conception de l'onglet Flux de contrôle.  
27) Bouton droit sur la nouvelle tâche de flux de données → **Renommer**  
28) Choisissez le nom « **ETC Currency Data** » (par exemple)  
29) Bouton droit sur la tâche de flux de données → **Propriétés**  
30) Dans la fenêtre Propriétés, vérifiez que la propriété **LocaleID** est définie à **Anglais (États-Unis)**.



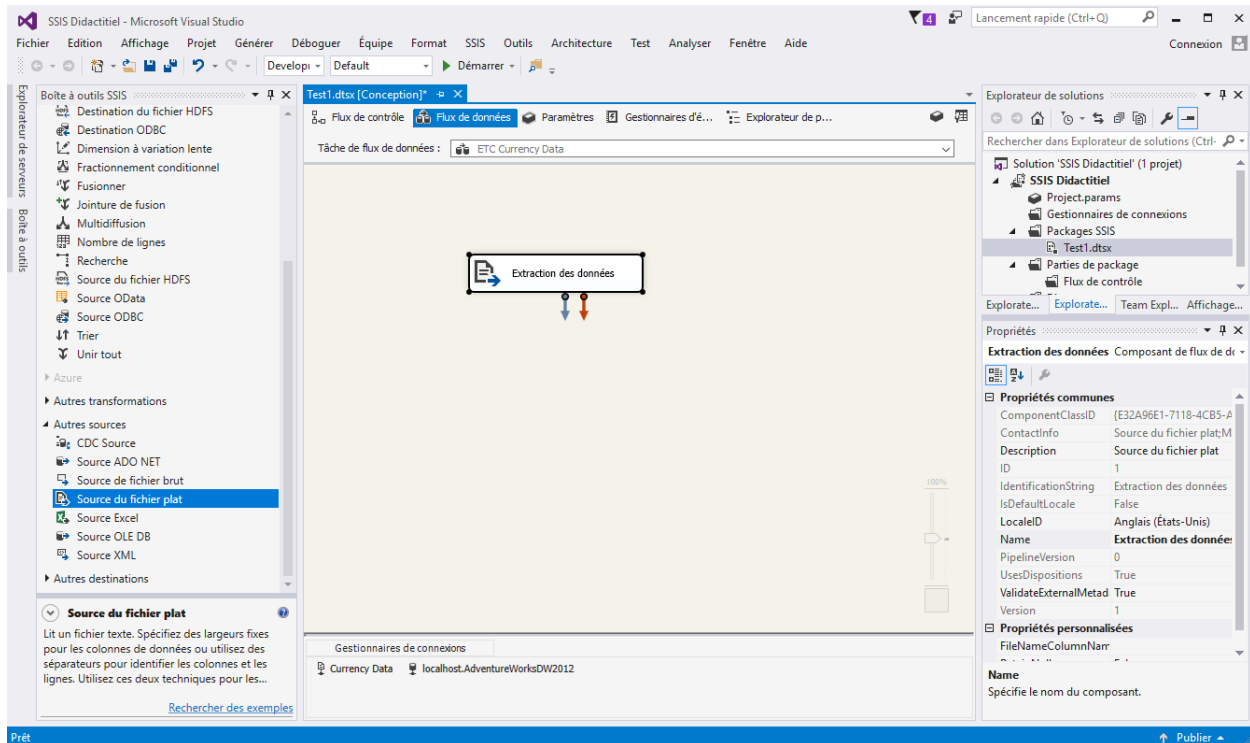
## Ajout et configuration de la source de fichier plat

Maintenant, on doit indiquer d'où proviennent les données pour notre nouvelle tâche de flux de données : on veut utiliser le fichier plat SampleCurrencyData.txt comme source, et on se servira d'une **source de fichier plat** et du gestionnaire de connexions de fichiers plats qu'on a créé pour le faire.

Une source de fichier plat est un composant de flux de données qui utilise des métadonnées définies par un Gestionnaire de connexions de fichiers plats pour spécifier le format et la structure des données à extraire.

### Pour ajouter un composant source de fichier plat :

- 31) Ouvrez le concepteur Flux de données en double-cliquant sur la tâche de flux de données « **ETC Currency Data** » ou en cliquant sur l'onglet Flux de données.
- 32) Dans la Boîte à outils, développez **Autres sources**, puis faites glisser **Source du fichier plat** dans la zone de conception de l'onglet Flux de données.
- 33) Bouton droit sur la nouvelle source du fichier plat → **Renommer**
- 34) Choisissez le nom « Extraction de données ».
- 35) Double-cliquez sur la source de fichier plat.
- 36) Dans la zone **Gestionnaire de connexions de fichiers plats**, tapez ou sélectionnez **Currency Data**.
- 37) Sélectionnez **Colonnes** et vérifiez si les noms des colonnes sont corrects.
- 38) Cliquez sur OK.
- 39) Bouton droit sur la source de fichier plat → **Propriétés**
- 40) Dans la fenêtre Propriétés, vérifiez que la propriété **LocaleID** est définie à **Anglais (États-Unis)**.



## Ajout et configuration des transformations de recherche (« lookups »)

Rappelons que dans la table qu'on veut charger, **FactCurrencyRate**, les colonnes **CurrencyKey** et **DateKey** sont des clés étrangères, dont les valeurs proviennent des tables de dimension **DimCurrency** et **DimDate**.

dbo.FactCurrencyRate
Columns
CurrencyKey (PK, FK, int, not null)
DateKey (PK, FK, int, not null)
AverageRate (float, not null)
EndOfDayRate (float, not null)
Date (datetime, null)
InsertDate (datetime, null)

Nos données sources ne contiennent pas ces clés. C'est à dire que « USD » doit être recherché dans la table **DimCurrency** pour trouver la valeur de la clé correspondant, appelée **CurrencyKey**, et chaque valeur de date doit être cherché dans la table **DimDate** pour trouver la clé **DateKey**.

- Pour obtenir **CurrencyKey**, on doit faire une jointure avec la table **DimCurrency** en utilisant la colonne **CurrencyAlternateKey**.
- Pour obtenir **DateKey**, on doit faire une jointure avec la table **DimDate** en utilisant la colonne **FullDateAlternateKey**.

Pour ce faire, on utilise une **transformation de recherche** (« Lookup »), capable d'effectuer une jointure pour chercher des données dans un dataset de référence.

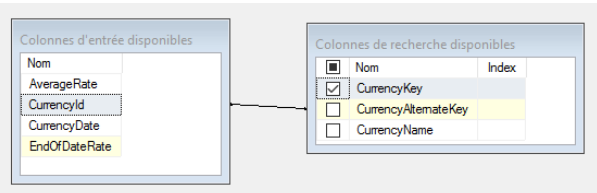
Ce dataset de référence peut être une table ou une vue existante, une nouvelle table, le résultat d'une instruction SQL ou même un cache qui contient le jeu de données de référence.

On utilisera deux transformations de recherche, une pour chaque colonne recherchée.

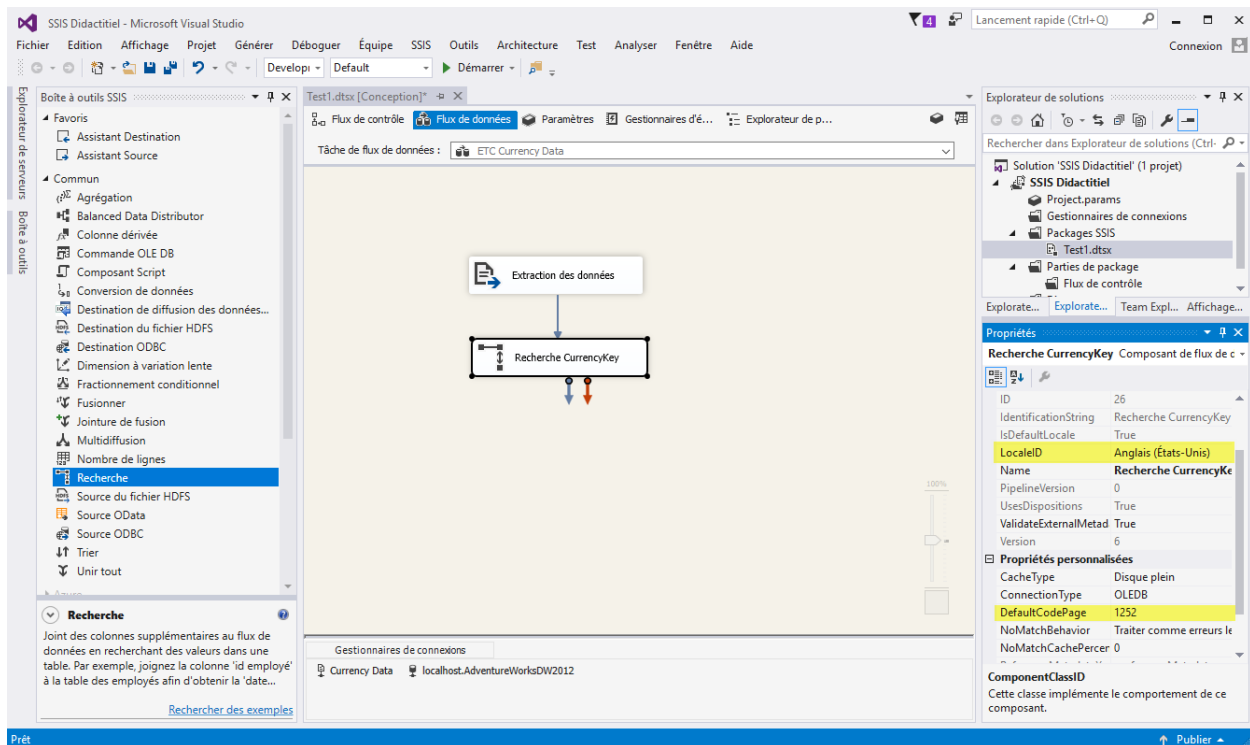
Dans les deux cas, les transformations de recherche utilisent le **Gestionnaire de connexions OLE DB** que vous avez créé précédemment.

*Pour ajouter et configurer la transformation Recherche CurrencyKey :*

- 41) Dans la Boîte à outils, développez **Commun**, puis faites glisser **Recherche** dans la zone de conception de l'onglet Flux de données.
- 42) Placez la recherche directement sous la source « Extraction de données ».
- 43) Sélectionnez la source de fichier plat « Extraction de données » et faites glisser la flèche bleue vers la transformation de recherche que vous venez d'ajouter pour connecter les deux composants.
- 44) Dans l'aire de conception Flux de données, cliquez sur **Recherche** dans la transformation de Recherche, puis remplacez le nom par « Recherche CurrencyKey »
- 45) Double-cliquez sur la transformation Recherche CurrencyKey.
- 46) Dans la page **Général**, effectuez les sélections suivantes :
  - a. Sélectionnez **Cache complet**.
  - b. Dans la zone Type de connexion, sélectionnez **Gestionnaire de connexions OLE DB**.
- 47) Dans la page **Connexion**, effectuez les sélections suivantes :
  - a. Assurez-vous que localhost.AdventureWorksDW2012 est affiché.
  - b. Sélectionnez « Utiliser une table ou vue »
  - c. Sélectionnez la table [dbo].[DimCurrency]
- 48) Dans la page Colonnes, effectuez les sélections suivantes :
  - a. Dans le volet Colonnes d'entrée disponibles, faites glisser **CurrencyID** vers le volet Colonnes de recherche disponibles et déposez cet élément sur **CurrencyAlternateKey**.
  - b. Dans la liste Colonnes de recherche disponibles, activez la case à cocher de **CurrencyKey**.

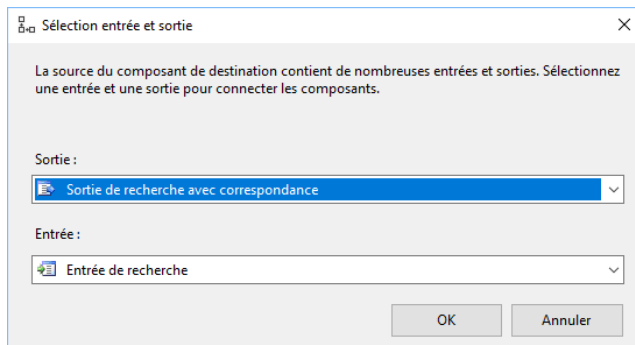


- 49) Cliquez sur OK pour revenir à l'aire de conception Flux de données.
- 50) Sélectionnez la transformation **Recherche CurrencyKey**.
- 51) Dans la fenêtre Propriétés, vérifiez que la propriété **LocaleID** a la valeur Anglais (États-Unis) et la propriété **DefaultCodePage** la valeur 1252.



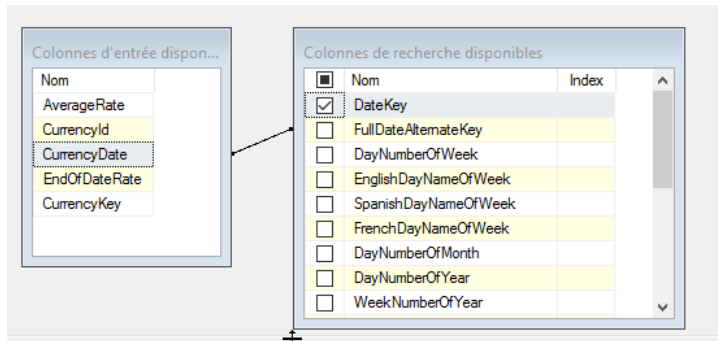
### *Pour ajouter et configurer la transformation Recherche TimeKey :*

- 52) Encore une fois, de la Boîte à outils, faites glisser **Recherche** dans la zone de conception de l'onglet Flux de données.
- 53) Placez la nouvelle recherche directement sous la transformation de recherche « Recherche CurrencyKey ».
- 54) Sélectionnez la transformation « Recherche CurrencyKey » et faites glisser la flèche bleue vers la nouvelle transformation de recherche que vous venez d'ajouter pour connecter les deux composants.
- 55) Dans la boîte de dialogue **Sélection entrée et sortie**, cliquez sur **Sortie de recherche avec correspondance** **avec correspondance** dans la zone de liste de Sortie
- 56) Cliquez sur OK.

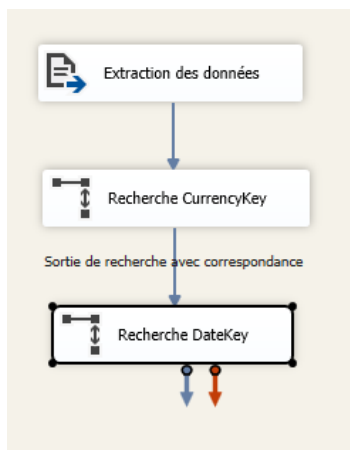


- 57) Remplacez le nom de la nouvelle transformation par « Recherche DateKey »
- 58) Double-cliquez sur la transformation Recherche DateKey.
- 59) Dans la page **Général**, effectuez les sélections suivantes :

- a. Sélectionnez **Cache partiel**.
  - b. Dans la zone Type de connexion, sélectionnez **Gestionnaire de connexions OLE DB**.
- 60) Dans la page **Connexion**, effectuez les sélections suivantes :
- a. Assurez-vous que localhost.AdventureWorks est affiché.
  - b. Sélectionnez « Utiliser une table ou vue »
  - c. Sélectionnez la table [dbo].[DimDate]
- 61) Dans la page Colonnes, effectuez les sélections suivantes :
- a. Dans le volet Colonnes d'entrée disponibles, faites glisser **CurrencyDate** vers le volet Colonnes de recherche disponibles et déposez cet élément sur **FullDateAlternateKey**.
  - b. Dans la liste Colonnes de recherche disponibles, activez la case à cocher de **DateKey**.



- 62) Dans la page Avancé, examinez les options de mise en cache.
- 63) Cliquez sur OK pour revenir à l'aire de conception Flux de données.
- 64) Sélectionnez la transformation Recherche DateKey, puis cliquez sur Propriétés.
- 65) Dans la fenêtre Propriétés, vérifiez que la propriété **LocaleID** a la valeur Anglais (États-Unis) et la propriété **DefaultCodePage** la valeur 1252.

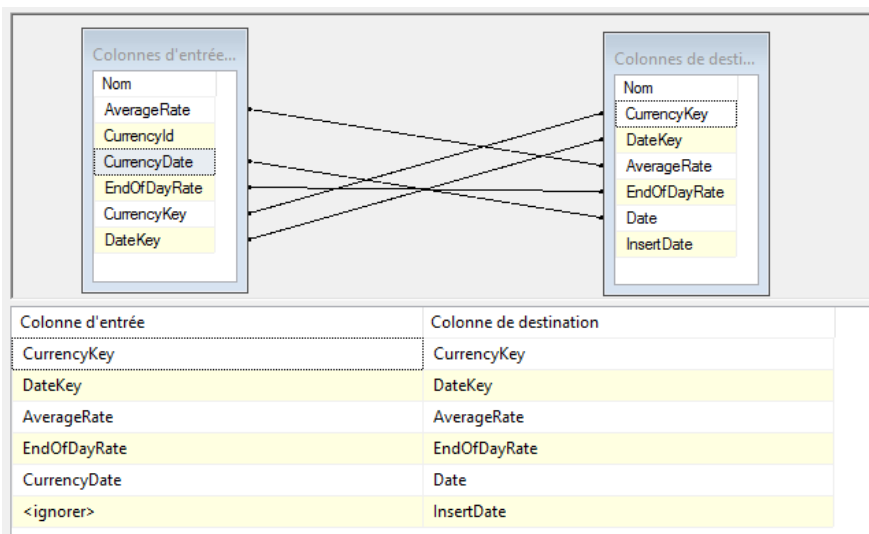


### Ajout et configuration de la destination OLE DB

Maintenant, on peut charger les données transformées dans la destination. Pour ce faire, on ajoutera une destination OLE DB au flux de données. Ce type de destination peut utiliser une table de base de données, un affichage ou une commande SQL pour charger les données dans plusieurs bases de données compatibles OLE DB.

- 66) Dans la Boîte à outils, développez **Autres Destinations**, puis faites glisser Destination OLE DB dans la zone de conception de l'onglet Flux de données.

- 67) Placez la destination OLE DB directement sous la transformation « Recherche DateKey ».
- 68) Sélectionnez la transformation Recherche DateKey et faites glisser la flèche bleue vers la nouvelle Destination OLE DB pour connecter les deux composants.
- 69) Dans la zone de liste de Sortie, cliquez sur **Sortie de recherche avec correspondance**
- 70) Cliquez sur OK.
- 71) Renommez la nouvelle destination à « Destination FactCurrencyRate ».
- 72) Double-cliquez sur la nouvelle destination.
- 73) Assurez-vous que localhost.AdventureWorksDW2012 est sélectionné dans la zone Gestionnaire de connexions OLE DB.
- 74) Dans la zone **Nom de la table ou de la vue**, tapez ou sélectionnez [dbo].[FactCurrencyRate].
- 75) Cliquez sur **Mappages**.
- 76) Assurez-vous que les colonnes d'entrée **AverageRate**, **CurrencyKey**, **EndOfDayRate**, **CurrencyDate** et **DateKey** sont correctement mappées aux colonnes de destination. (Si des colonnes aux noms identiques sont mappées, le mappage est correct).



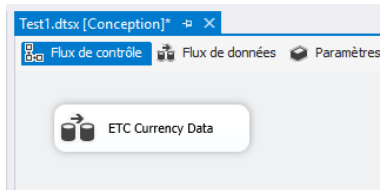
- 77) Cliquez sur OK.
- 78) Cliquez avec le bouton droit sur la Destination OLE DB exemple, puis cliquez sur **Propriétés**.
- 79) Dans la fenêtre Propriétés, vérifiez que la propriété **LocaleID** est définie à Anglais (États-Unis) et la propriété **DefaultCodePage** à 1252.
- 80) Enregistrer le package.



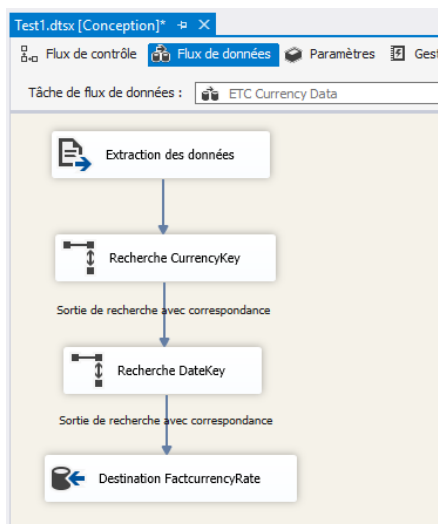
## Exécuter / déboguer le package

On est maintenant prêt à exécuter le package.

Dans l'onglet **Flux de contrôle**, vous devriez avoir un seul flux de données :



Et dans l'onglet **Flux de données**, vous aurez tous les détails de ce flux de données :



Pour vous assurer que tout est correct, regardez le contenu de la table FactCurrencyKey avant d'exécuter le package (dans SQL Server Management Studio). Par exemple :

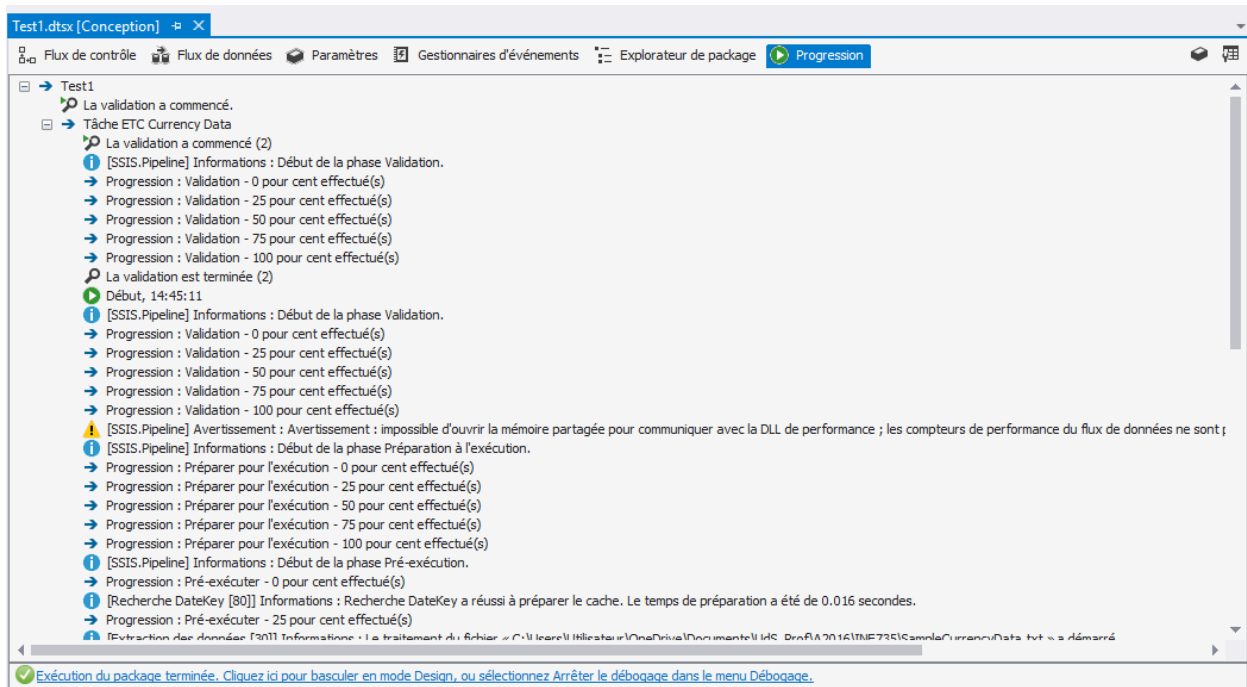
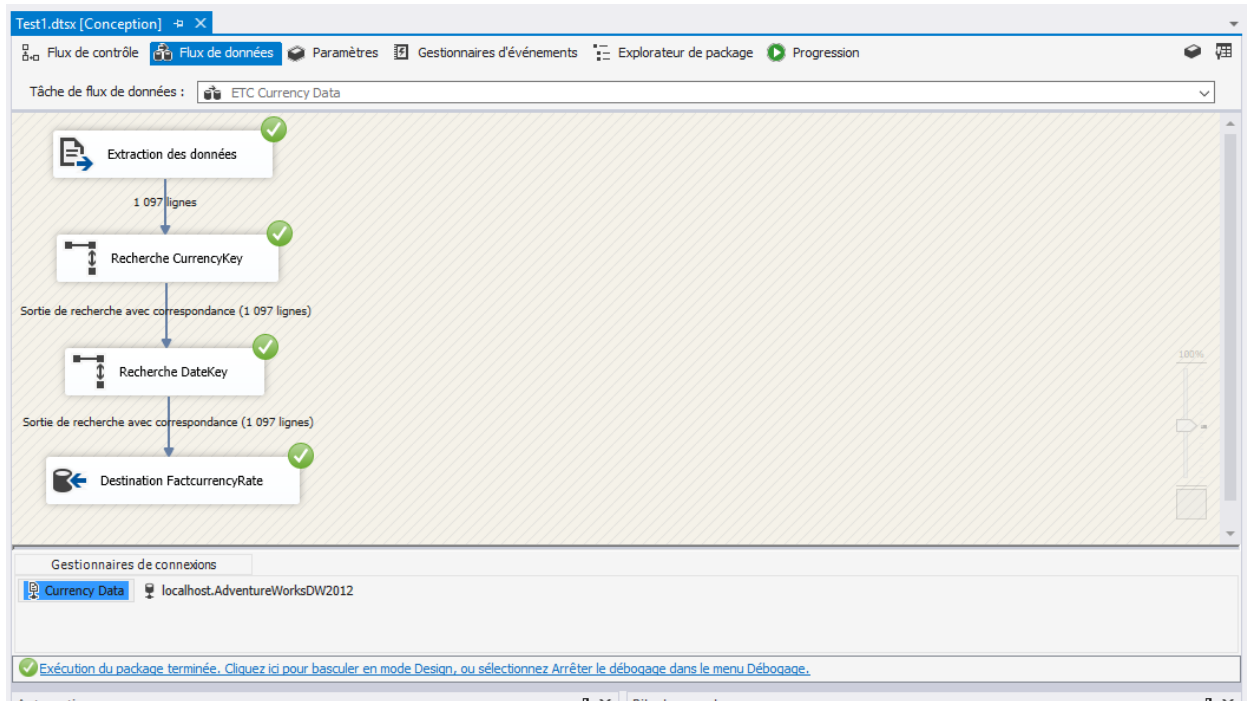
```
SELECT COUNT (*) FROM FactCurrencyRate
```

Remarquez aussi que la colonne InsertDate, qu'on vient d'ajouter, est nulle pour les enregistrements qui existaient déjà dans la base de données. Cette nouvelle colonne aura automatiquement la valeur de la date actuelle au moment de l'insertion, pour les nouvelles données qu'on va insérer.

### Pour exécuter le nouveau package :

81) Dans le menu Déboguer, cliquez sur Démarrer le débogage.

SSIS montrera l'état d'avancement du processus dans l'onglet Flux de données et dans l'onglet Résultat d'exécution :



82) Le package s'exécute et 1 097 lignes sont ajoutées à la table de faits FactCurrencyRate dans l'entrepôt de données AdventureWorksDW2012.

Vous pouvez regarder les nouvelles lignes avec la commande SQL :

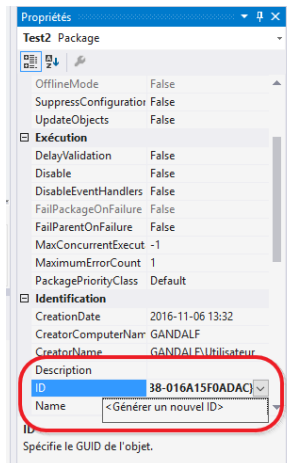
```
SELECT *  
FROM vFactCurrencyRate f  
WHERE InsertDate is not null
```

83) Une fois l'exécution du package terminée, dans le menu Déboguer, cliquez sur **Arrêter le débogage**.

## Copie d'un package SSIS

Vous pouvez facilement copier un package SSIS pour le modifier tout en conservant la version précédente.

- 1) Dans l'Explorateur de solutions : bouton droit sur le package que vous désirez copier → **Copier**.
- 2) Dans l'Explorateur de solutions : bouton droit sur Packages SSIS → **Coller**.
- 3) Double-cliquez sur le nouveau package qui a été créé pour l'ouvrir.
- 4) Cliquez n'importe où dans l'arrière-plan de l'onglet Flux de contrôle
- 5) Bouton droit → **Propriétés**.
- 6) Dans la fenêtre Propriétés, mettre la propriété **Name** à jour.
- 7) Cliquez sur la propriété **ID**, puis dans la liste, cliquez sur **<Générer un nouvel ID>**.



8) Voilà, votre nouveau package est prête à utiliser !

## Redirection de flux d'erreurs

Pour traiter les erreurs qui risquent de se produire dans le processus de transformation, SSIS vous permet de décider par composant et par colonne comment traiter les données qui ne peuvent pas être traités.

Vous pouvez choisir de :

- Ignorer une erreur dans certaines colonnes
- Rediriger dans sa totalité la ligne qui a échoué, ou
- Simplement, de faire échouer le composant.

Par défaut, tous les composants sont configurés pour **échouer lorsque des erreurs se produisent**. Ce fait entraîne l'échec du package et l'arrêt de tous les traitements ultérieurs.

Mais souvent, on préfère de configurer et de traiter les erreurs au moment où elles se produisent. Vous pouvez choisir d'ignorer les erreurs pour permettre que le package s'exécute au complet, mais en redirigeant la ligne qui a échoué vers un autre chemin de traitement où les données et l'erreur peuvent être rendues persistantes, étudiées et retraitées ultérieurement.

Dans cette section, on créera une version endommagée d'un fichier d'exemple, ce qui entraînera une erreur de traitement. Pour gérer les données d'erreur, vous allez ajouter et configurer une **destination de fichier plat** chargée d'écrire dans un fichier toutes les lignes qui ne parviennent pas à détecter une valeur de recherche dans la transformation « Recherche CurrencyKey ».

### Création d'un fichier endommagé :

- 1) Dans le Bloc-notes ou un autre éditeur de texte, ouvrez le fichier **Currency\_VEB.txt**.
- 2) Remplacer quelques instances de « VEB » par « BAD ». Cette valeur, évidemment, n'existe pas dans la table DimCurrency.
- 3) Enregistrez le fichier modifié sous le nom Currency\_BAD.txt.

### Copie du package :

- 4) Si vous ne l'avez pas fait déjà, créez une copie du package pour la modifier.

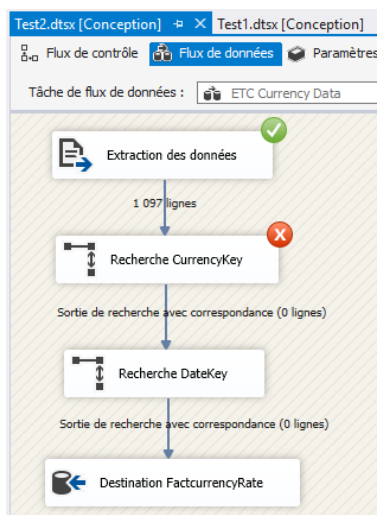
### Changement du fichier source :

- 5) Dans le nouveau package, double-cliquez sur le gestionnaire de connexions de fichier plat « Currency Data » pour le modifier.
- 6) Dans la page Générale, modifiez le nom du fichier en cliquant sur Parcourir... et en sélectionnant le fichier Currency\_BAD.txt.

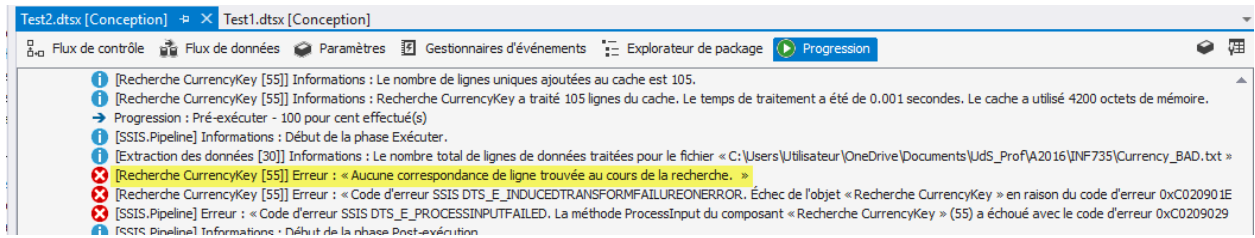
### Test du package

- 7) Essayez d'exécuter le package (menu Déboguer → Démarrer le débogage)

Le package échoue, et la transformation Recherche CurrencyKey est marquée en rouge :



- 8) Dans l'onglet Résultats d'exécution, remarquez que la transformation Recherche CurrencyKey a échoué avec le message d'erreur « Row yielded no match during lookup ».



9) Arrêtez le débogage.

## Ajout de redirection de flux d'erreurs

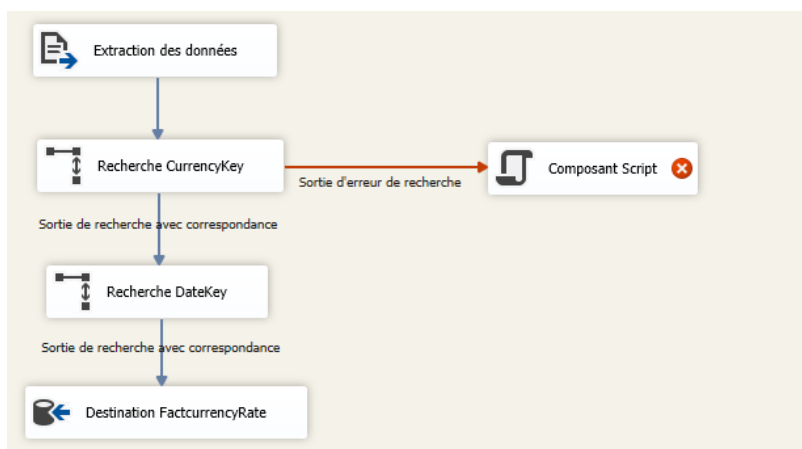
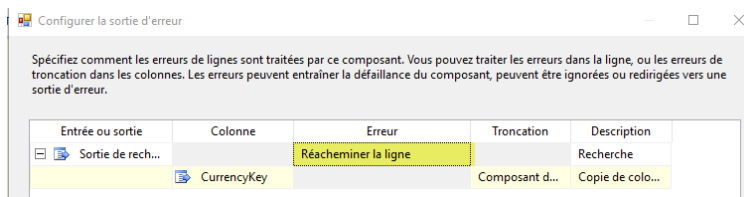
Maintenant, on modifiera la transformation Recherche CurrencyKey afin qu'elle réachemine toutes les lignes échouant lors de la sortie d'erreur. Ces lignes seront enregistrées dans un fichier.

SSIS offre deux colonnes supplémentaires pour les sorties d'erreur : **ErrorCode** et **ErrorColumn**. Mais elles ne contiennent que des codes numériques identifiant un numéro d'erreur et l'ID de la colonne, ce qui n'est pas très utile sans la description d'erreur correspondante.

On utilisera alors un composant **Script** pour accéder à l'API de SSIS et obtenir une description de l'erreur avant que le package n'enregistre les lignes ayant échoué dans le fichier.

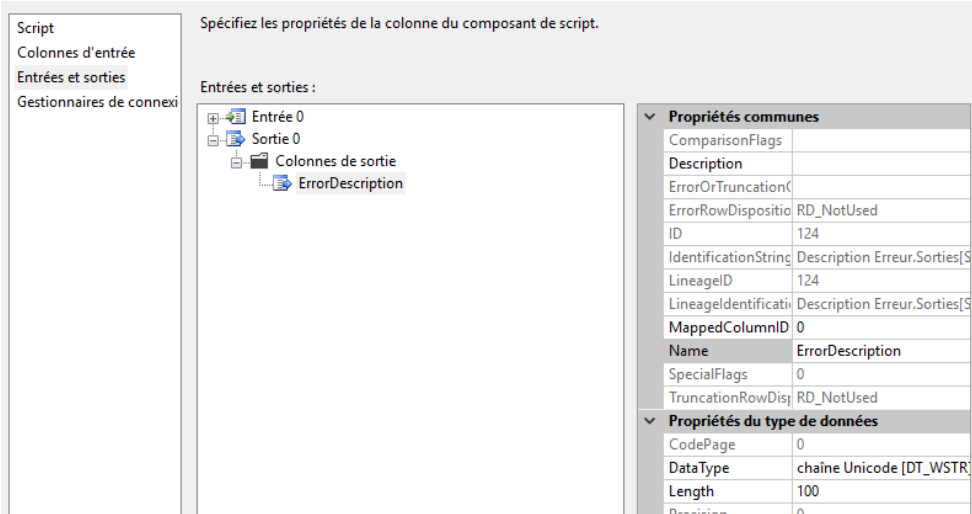
### Pour configurer une sortie d'erreur :

- 10) Dans la Boîte à outils, développez **Commun**, puis faites glisser l'option **Composant Script** dans le Flux de données.
- 11) Placez le composant Script à droite de la transformation Recherche CurrencyKey.
- 12) Dans Sélectionner le type de composant de script, cliquez sur **Transformation**, puis sur OK.
- 13) Cliquez sur la transformation Recherche Currency Key et faites glisser la **flèche rouge** vers la nouvelle transformation Script.
- 14) Dans la colonne Erreur, sélectionnez **Réacheminer la ligne**, puis OK.



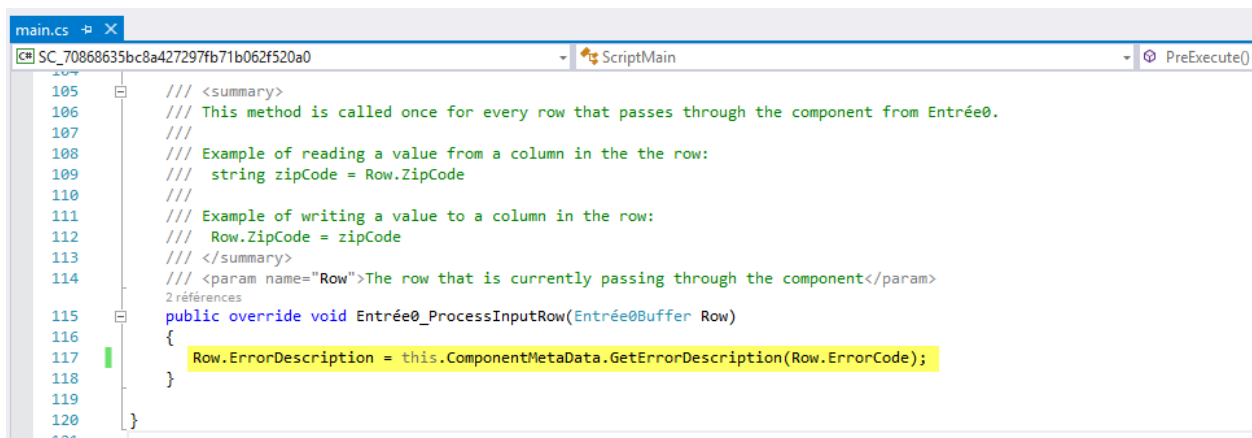
La flèche rouge représente la sortie d'erreur de la transformation Recherche CurrencyKey.

- 15) Changez le nom du nouveau composant Script par « **Description Erreur** ».
- 16) Ouvrez cette nouvelle transformation.
- 17) Dans la page **Colonnes d'entrée**, cochez la colonne **ErrorCode**.
- 18) Dans la page **Entrées et sorties**, développez **Sortie 0**, cliquez sur **Colonnes de sortie**, puis sur **Ajouter une colonne**.
- 19) Dans la propriété **Name**, tapez **ErrorDescription**
- 20) Dans la propriété **DataType**, attribuez la valeur string [DT\_WSTR], avec une longueur de 100.



- 21) Dans la page **Script**, vérifiez que la propriété **LocaleID** est définie à Anglais (États-Unis).
- 22) Cliquez sur le bouton **Modifier le script** (en bas de la boîte de dialogue) pour ouvrir Visual Studio Tools for Applications (VSTA).
- 23) Dans la méthode **Entrée0\_ProcessInputRow**, tapez le code suivant :
  - [Visual Basic] : Me.ComponentMetaData.GetErrorDescription(Row.ErrorCode)
  - [Visual C#] : Row.ErrorDescription = this.ComponentMetaData.GetErrorDescription(Row.ErrorCode);

Exemple:



- 24) Enregistrez le script et fermez la fenêtre VSTA.
- 25) Cliquez sur OK pour fermer la boîte de dialogue **Éditeur de transformation de script**.

Cette transformation ajoute la colonne **ErrorDescription** à la liste de colonnes de sortie du Script, nous permettant de l'utiliser plus tard.

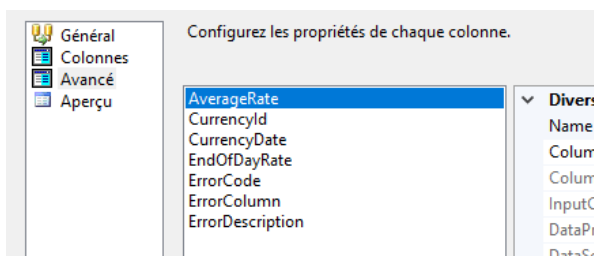
### Ajout d'une destination de fichier plat

Maintenant, on enregistrera toutes les informations sur les lignes échouées dans un fichier délimité pour les traiter ultérieurement.

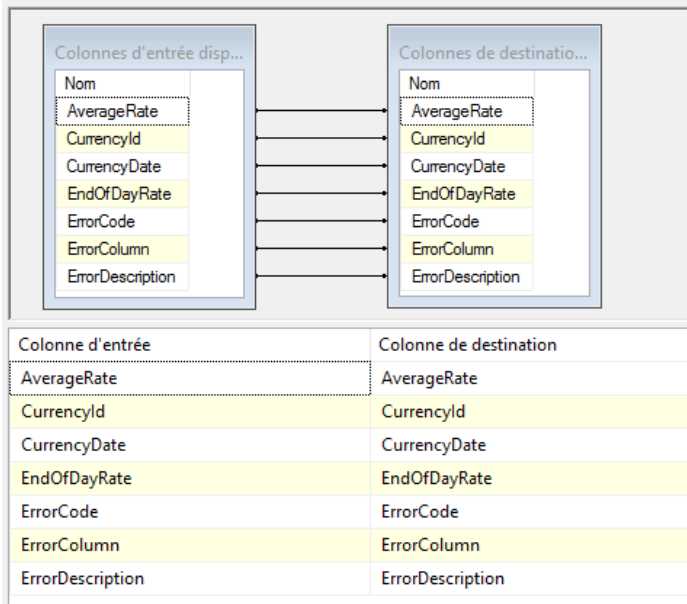
Pour ce faire, vous devez ajouter et configurer un gestionnaire de connexions de fichiers plats pour le fichier d'erreur, ainsi qu'une destination de fichier plat.

- 26) Dans la Boîte à outils, développez **Autres destinations**, puis faites glisser **Destination de fichier plat** vers le flux de données.
- 27) Placez la nouvelle destination de fichier plat directement sous la transformation Description Erreur.
- 28) Sélectionnez la transformation Description Erreur, puis faites glisser la flèche bleue vers la nouvelle destination.
- 29) Renommez la nouvelle destination à « **Erreurs CurrencyKey** ».
- 30) Ouvrez la nouvelle destination, puis cliquez sur **Nouveau** pour créer un nouveau gestionnaire de connexion de fichier plat.
- 31) Assurez-vous que Délimité est sélectionné, puis cliquez sur OK.
- 32) Dans la zone Nom du gestionnaire de connexions, tapez « Données d'erreur ».
- 33) Cliquez sur Parcourir, puis sélectionnez le dossier dans lequel le fichier est à stocker.
- 34) Dans la zone Nom de fichier, tapez **SortieErreur.txt**, puis cliquez sur **Ouvrir**.
- 35) Vérifiez que la zone **Paramètres régionaux** contient la valeur Anglais (États-Unis) et la zone **Page de codes** la valeur 1252 (ANSI - Latin I).
- 36) Sélectionnez le volet Avancé.

Notez qu'en plus des colonnes du fichier de données source, trois nouvelles colonnes sont présentes : **ErrorCode**, **ErrorColumn** et **ErrorDescription**. Ces colonnes sont générées par la sortie d'erreur de la transformation Recherche CurrencyKey et par le script. Vous pouvez les utiliser pour résoudre le problème à l'origine de l'échec de la ligne.

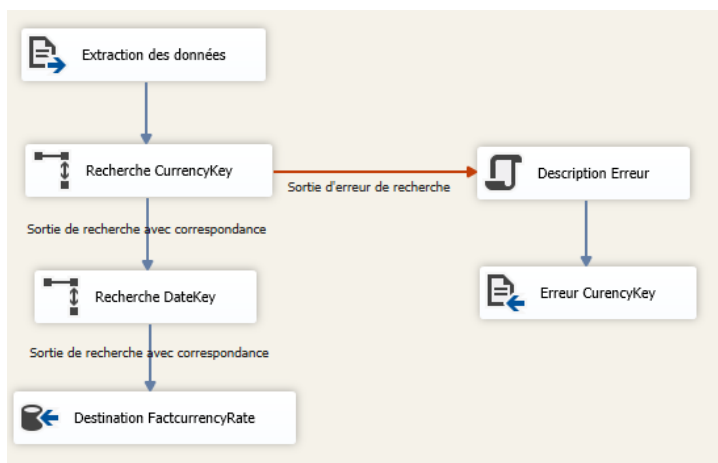


- 37) Cliquez sur OK.
- 38) Dans l'Éditeur de destination de fichier plat, désactivez la case à cocher **Remplacer les données du fichier** pour conserver les erreurs sur plusieurs exécutions de packages.
- 39) Cliquez sur Mappages pour vérifier que toutes les colonnes sont correctes. Vous pouvez également renommer les colonnes dans la destination.



40) Cliquez sur OK.

41) Enregistrer le package, il ressemble à ceci maintenant :



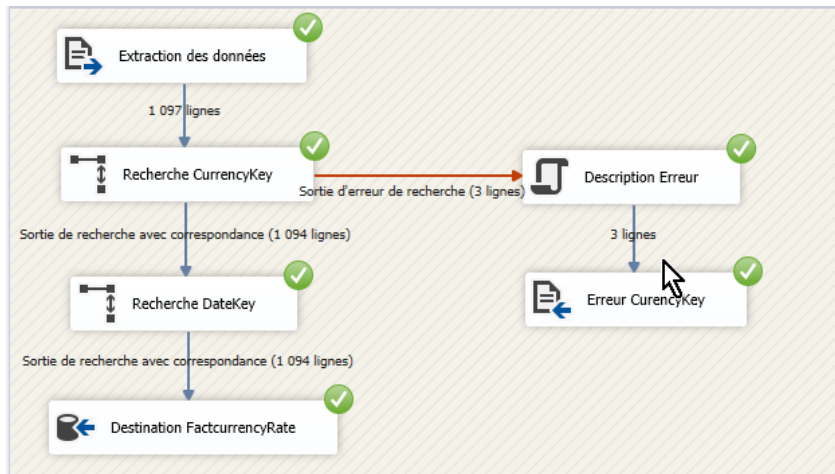
## Test du package

Maintenant vous pouvez exécuter le package (Déboguer → Démarrer le débogage)

Vérifiez que :

- Le package s'exécute correctement.
- Certaines lignes ont été redirigées car des erreurs ont été trouvées (le nombre de lignes peut être différent dans votre cas, selon combien de lignes vous avez modifiées) :





- Ces lignes se retrouvent dans le fichier « SortieErreur.txt », avec la description de l'erreur survenue. Par exemple :

```
SortieErreur.txt - Bloc-notes
Fichier  Edition  Format  Affichage  ?
0.00158,BAD,2005-01-02,0.00158,-1071607778,0,Aucune correspondance de ligne trouvée au cours de la recherche.
0.00157,BAD,2005-01-08,0.00157,-1071607778,0,Aucune correspondance de ligne trouvée au cours de la recherche.
0.00157,BAD,2005-01-20,0.00157,-1071607778,0,Aucune correspondance de ligne trouvée au cours de la recherche.
```

- N'oubliez pas d'arrêter le débogage.

## Autres didacticiels intéressants :

On vous suggère de regarder rapidement les tutoriels suivants pour plus d'information :

- Ajout de configurations à un package : <http://technet.microsoft.com/fr-fr/library/ms167153.aspx>
- Ajout du mode d'écriture dans un journal : <http://technet.microsoft.com/fr-fr/library/ms167061.aspx>
- Ajout d'un bouclage : <http://technet.microsoft.com/fr-fr/library/ms166566.aspx>
- Vidéo (création d'un package simple) [http://msdn.microsoft.com/fr-fr/library/cc952921\(v=SQL.100\).aspx](http://msdn.microsoft.com/fr-fr/library/cc952921(v=SQL.100).aspx)
- Création d'un lookup pour faire INSERT ou UPDATE : <http://www.insideit.fr/post/2010/06/17/Lookup-simple-sur-SSIS>

Vous trouverez aussi de nombreux packages d'exemple dans **C:\Program Files\Microsoft SQL Server\100\Samples\Integration Services\Package Samples** (pour installer les packages d'exemple, allez à <http://msftdbprodsamples.codeplex.com>)