

Talend Developer Documentation

Last updated by | Archana Balachandran | Jun 4, 2020 at 2:28 PM EDT

Contents

- [Staging Orchestration Process](#)
 - [Overview](#)
 - [Process Flow Diagram](#)
 - [Staging Master Orchestration process](#)
 - [Staging Control Orchestration](#)
 - [Individual Process Jobs](#)
 - [Major Components Used](#)
- [Staging Flat files form Azure Blob Storage](#)
 - [Overview:](#)
 - [Connect to Azure Storage and View files in Blob Contain...](#)
 - [Creating Talend Jobs to Load Flat Files](#)
 - [A. Adding Components and Context Variables](#)
 - [B. Configure each Talend component](#)
 - [C. Complete your Talend Job](#)
- [SSIS Package to SnowSQL Conversion](#)
 - [1. Download and View Latest SSIS Package in Visual Stu...](#)
 - [2. View SQL Code for Stored Procedure in SQL Server M...](#)
 - [3. Convert the Transact SQL Code to SnowSQL in Snowf...](#)
 - [4. Update SnowSQL scripts in Git Repository](#)
 - [NOTES AND HELPFUL RESOURCES:](#)

Staging Orchestration Process

Overview

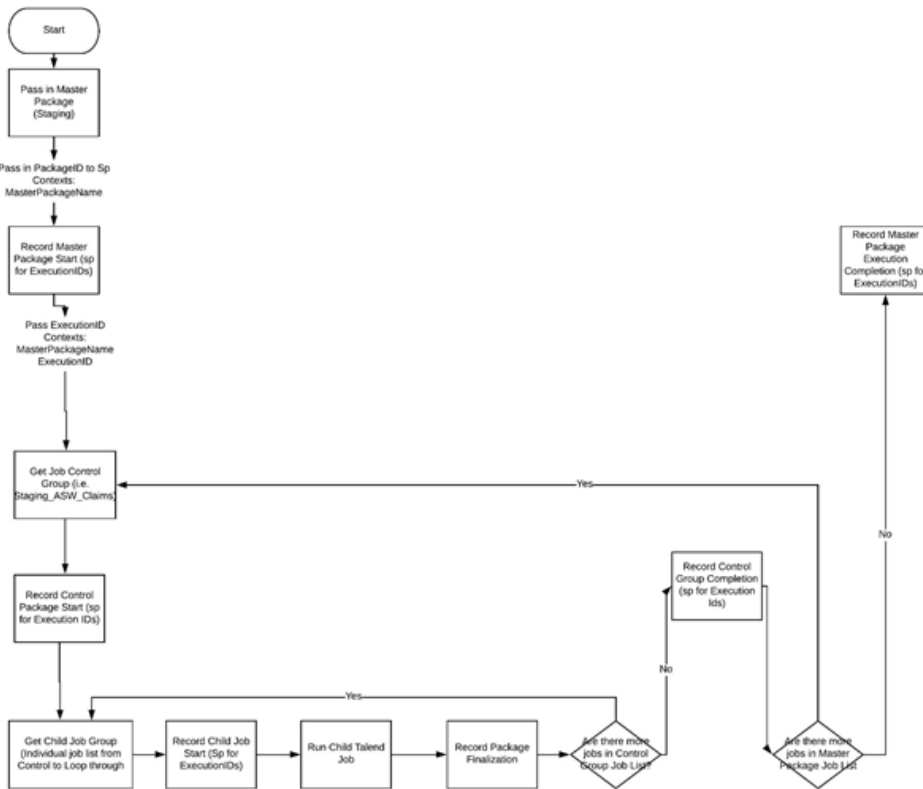
The orchestration of the daily process is achieved via Talend in the form of stages:

1. Staging Master Orchestration
2. Staging Control Orchestration

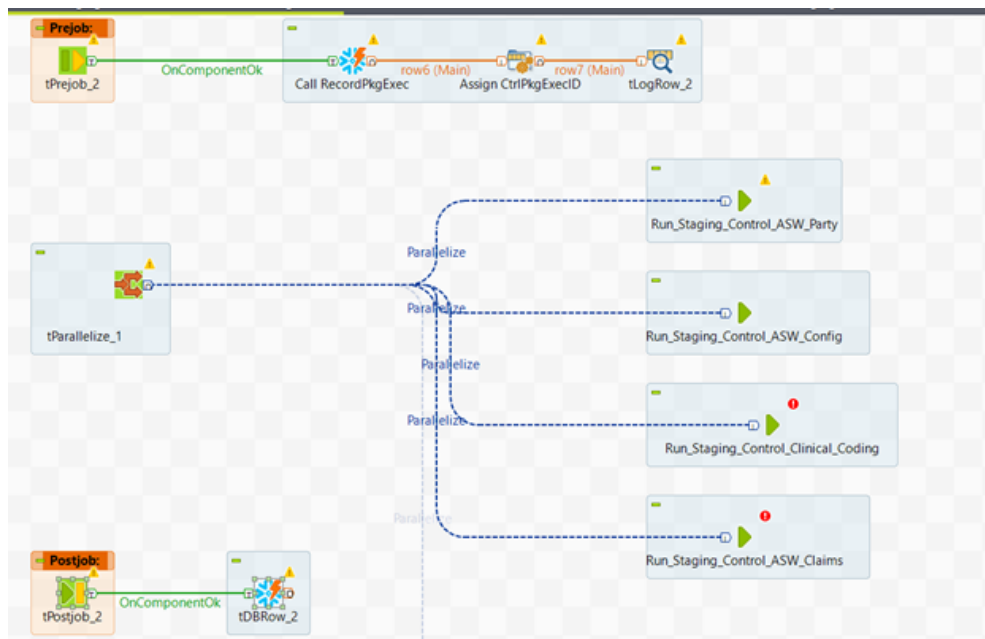
Each stage is explained in detail in the sections below.

Process Flow Diagram

The following diagram depicts the process for Staging Orchestration process.



Staging Master Orchestration process



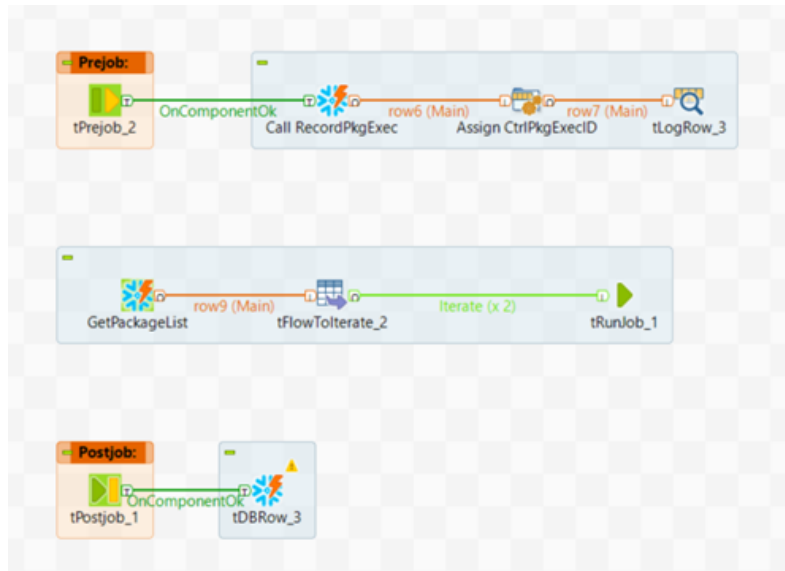
Staging Master Orchestration is responsible for executing the Control Jobs in parallel. Control Jobs are segregated based on schema/subject area. The tParallelize component is used to achieve maximum parallelism.

The PreJob component receives the Staging Master job name from the parent job and executes the `sp_recordpackageExecution` stored procedure in Snowflake. This stored procedure takes Job name, and Parent Execution ID as input arguments and returns the executionID as output. This value is then passed to the next subjob (Staging Process Orchestration) as a parameter. The parameters passed to the next subjobs are Job ID of Control Job, Job Name of Control Job and Execution ID of parent job (which is the Staging Master job).

Once all the Control Jobs return a successful status to this job, the PostJob component triggers the stored procedure to record Package

success by stamping an end time in the PackageExecution table for the ExecutionID.

Staging Control Orchestration

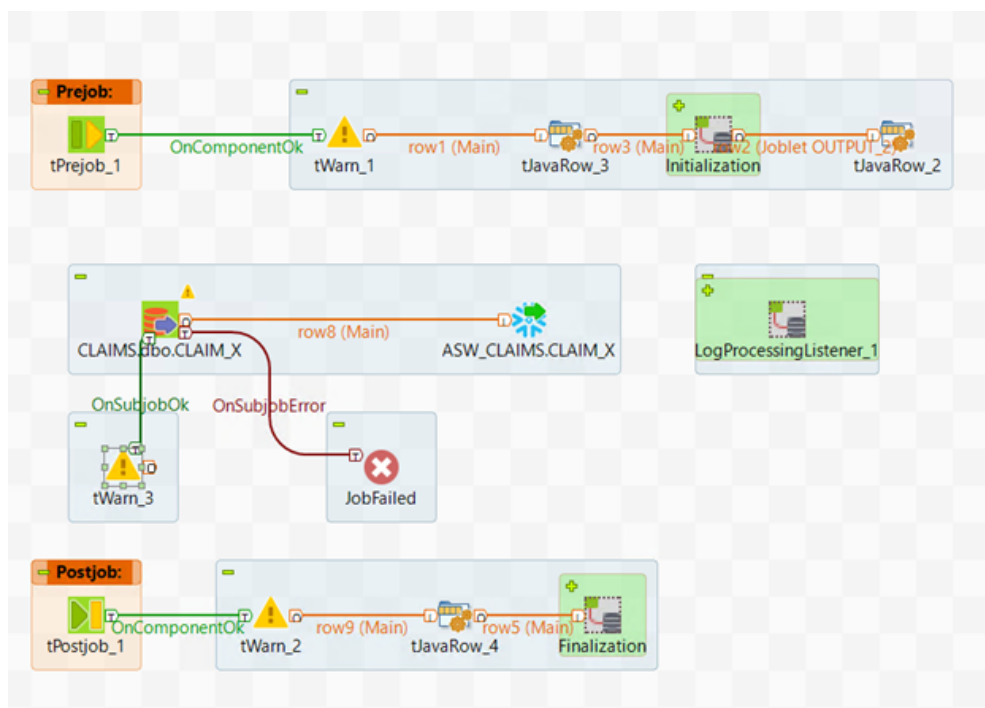


Staging Control Orchestration is responsible for invoking the process jobs. The Pre-job component uses the parameters Job name and Parent Package Execution ID from the parent job and computes the Execution ID.

Using the Job ID of Control Job which is passed as a parameter from the Parent Job, the Package table in ETLConfig database is queried to get the list of process jobs to run. Each package name is then passed to tFlowtolterate component, which iterates through each package name in the list and executes them by passing it to tRunJob component. tFlowtolterate component is configured to execute two process jobs in parallel.

Once the process jobs have successfully executed, the PostJob component executes the stored procedure to record the completion of package execution(sp_recordPackageSuccess). It passes the ExecutionID of the current job as input parameter, which is used to find the corresponding record in the PackageExecution table in ETLConfig database, and stamp an End time.

Individual Process Jobs



Process jobs are the most granular job executions in the daily process. These jobs are invoked by the Staging Process Orchestration job. This job performs one unit for work, for example, stages data from a SQL Server table to a Snowflake table. The PreJob Component executes the stored procedure in Snowflake to log the start of job execution by passing in the name of job, along with Parent Job Execution ID as parameters. This returns the execution ID of the current job as output.

Once the main job is complete, the successful execution is recorded by calling the Snowflake Stored procedure and passing the Execution ID of the current job as input parameter.

Major Components Used

- PostJob – used for closing connections and recording completion of package execution
- PreJob- used for validating connections, recording package executions
- tParallelize – used to execute multiple control jobs in parallel threads.
- tFlowtolterate – iterates through list of packagenames passed as input and passes them to tRunJob
- tDBRow – used to call Snowflake Stored Procedure, mostly for package execution logging.
- tRunJob – used to specify the child jobs to execute, and also specify the parameters to be passed to child jobs.
- Initialization Joblet - reusable job attached to PostJob component in each job that performs all the Initialization activities such as recording package execution status.
- Finalization Joblet – reusable job attached to PostJob component in each job that performs all the finalization activities such as recording package execution status
- LogProcessingListener Joblet – reusable code that listens to the execution of each job that it is attached to, and logs Java exceptions and any other exceptions reported by the tDie and tWarn components.
- tWarn – component used to allow LogListener joblet to catch any success messages or warnings from jobs in a controlled manner.
- tDie – component used to control the exit point in the code, by returning the error code that will be used by packageExecutionStatus.

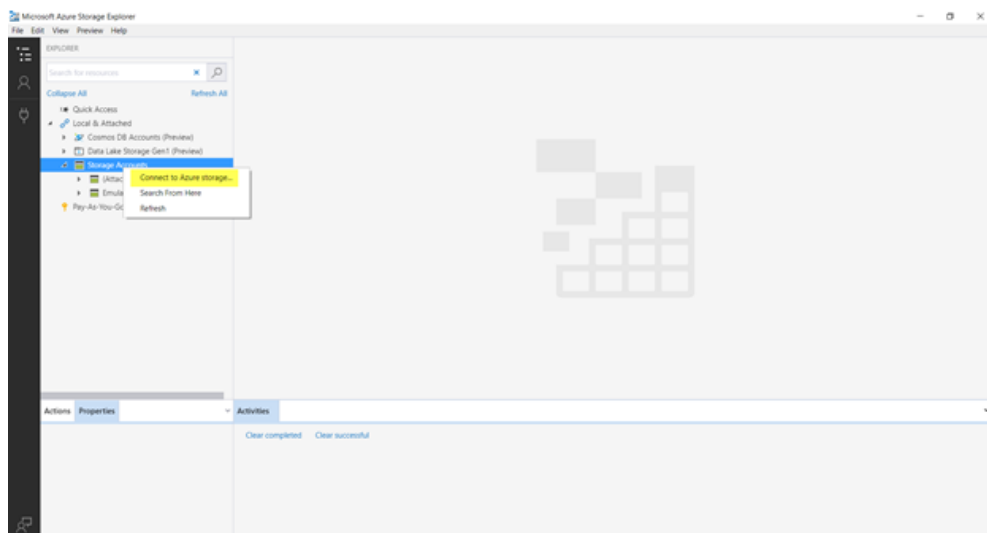
Staging Flat files form Azure Blob Storage

Overview:

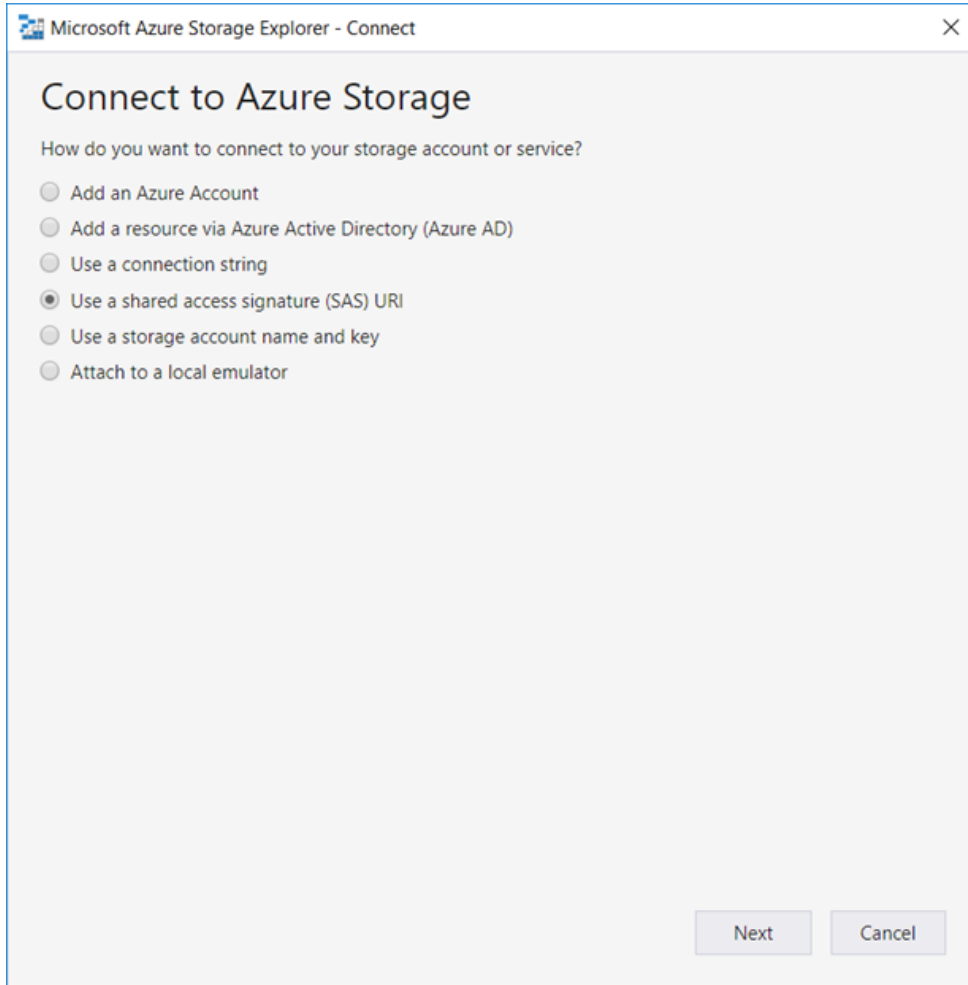
The purpose of this document is to list the various steps needed to successfully load the legacy files (IS4East and IS4West) from Azure Storage to Snowflake. The document also covers all the dependencies required to complete the task.


Connect to Azure Storage and View files in Blob Containers

1. To connect to Azure Storage, you need to download Microsoft Azure Storage Explorer (Free) for your version of OS (link below).
<https://azure.microsoft.com/en-us/features/storage-explorer/>
2. Open Microsoft Azure Storage Explorer, right click 'Storage Accounts' and select Connect to Azure Storage'.



3. In the Connection setup window, select 'Use a shared access Signature(SAS) URI'



4. In the next section, paste the SAS URI (given below) into the 'URI' field. All other fields will be automatically populated.
<https://covuseedwdevsa.blob.core.windows.net/?sv=2018-03-28&ss=bfqt&srt=sco&sp=rwdlacup&se=2019-12-08T04:15:15Z&st=2019-08-14T19:15:15Z&spr=https&sig=V1xOn8JjkbQQA4JNXLTaaxS8aLjasbCTiQtIxrnds0k%3D> 

Microsoft Azure Storage Explorer - Connect

Attach with SAS URI

Display name:

covuseedwdevsa

URI:

<https://covuseedwdevsa.blob.core.windows.net/?sv=2018-03-28&ss=bfqt&srt=sco&sp=rwdlacup&se=20>

Blob endpoint:

<https://covuseedwdevsa.blob.core.windows.net>

File endpoint:

<https://covuseedwdevsa.file.core.windows.net>

Queue endpoint:

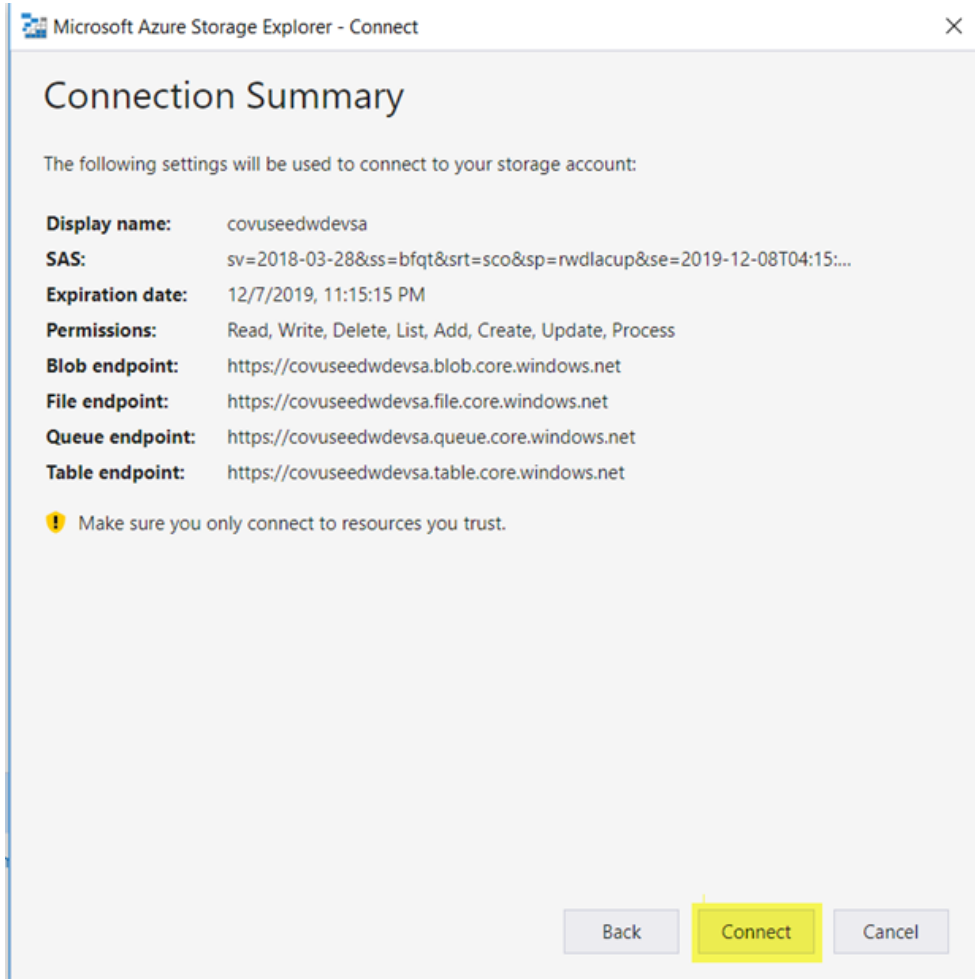
<https://covuseedwdevsa.queue.core.windows.net>

Table endpoint:

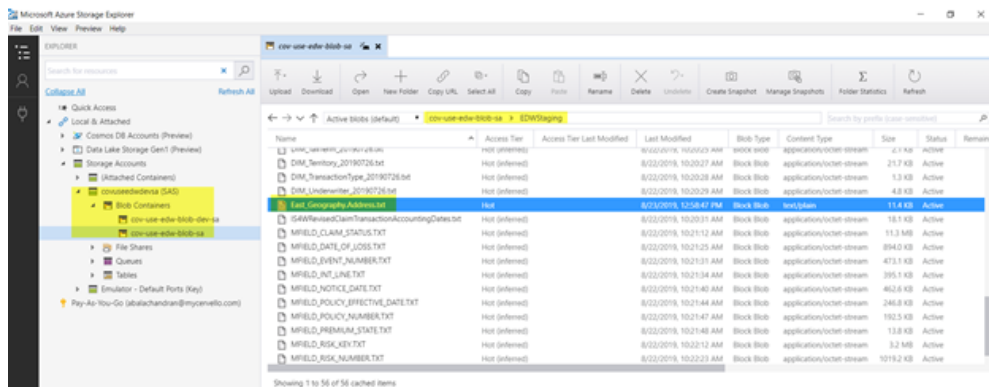
<https://covuseedwdevsa.table.core.windows.net>

Back Next Cancel

5. In the Connection Summary window, verify the details and click 'Connect'. Note: confirm you are connecting to the dev container by confirming the keyword 'dev' in the URI.



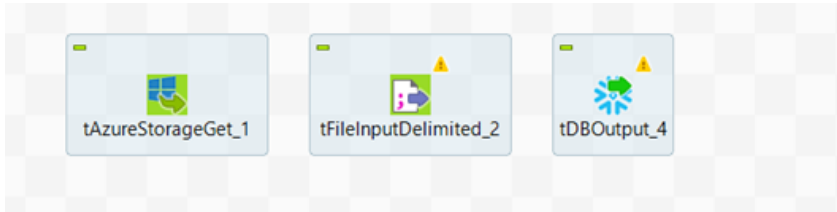
6. Once connection is successful, you will be able to view the new Storage Account and the associated containers and Blobs.



Creating Talend Jobs to Load Flat Files

A. Adding Components and Context Variables

1. Add the required components from Palette into Designer.
 - a. tAzureStorageGet – downloads files from specified Blob to your local directory
 - b. tFileInputDelimited – configures the flat file and prepares it for loading
 - c. tSnowflakeOutput – loads file configured in prior component



2. Add the necessary Context Variables for the job as shown below:

	Name	Type	Comment	Default Value
11	Conn_Snowflake_Staging_IS4EAST_role	String		DEV_ETL_ROLE
12	Conn_Snowflake_Staging_IS4EAST_db	String		DEV_STAGING
13	azurestoragecontainer	String		"cov-use-edw-blob-sa"
14	azurestorageblobprefix	String		"EDWStaging/East_Geography.Address.txt"
15	azurestorageasurl	String		"https://covuseedwdevsa.blob.core.windows.net/?sv=2018-03-28&ss=bfqt&srt=sco&sp=rwdlacup&se=2019-12-08T04:15:15Z&st=2019-08-14T19:15:15Z&spr=https&sig=V1xOn8JkbQQA4JNXLTaaxS8aLjasbCTiQtlxrnds0k%3D"
16	azurestagelocaldirectory	String		"C:/Users/Archana Balachandran/Desktop/Coverys/IS4LegacyFiles"
17	snowflakeoutboundlocaldir	String		"C:/Users/Archana Balachandran/Desktop/Coverys/IS4LegacyFiles/EDWStaging/East_Geography.Address.txt"
18	stagingexecutionid	int Integer		8888
19	extractionexecutionid	int Integer		8888

Context variable	Type	Value
azurestoragecontainer	String	cov-use-edw-blob-sa
azurestorageblobprefix	String	EDWStaging/East_Geography.Address.txt
azurestorageasurl	String	https://covuseedwdevsa.blob.core.windows.net/?sv=2018-03-28&ss=bfqt&srt=sco&sp=rwdlacup&se=2019-12-08T04:15:15Z&st=2019-08-14T19:15:15Z&spr=https&sig=V1xOn8JkbQQA4JNXLTaaxS8aLjasbCTiQtlxrnds0k%3D
azurestagelocaldirectory	String	<path_to_your_local_directory>\\IS4LegacyFiles
snowflakeoutboundlocaldir	String	<path_to_your_local_directory>/IS4LegacyFiles/EDWStaging/East_Geography.Address.txt
stagingexecutionid	Integer	8888
extractionexecutionid	Integer	8888

Note: Ensure that all strings are encapsulated in double quotes, and note the Type for ExecutionID columns. For the purpose of this tutorial, a folder named 'IS4LegacyFiles' was created in the local machine, and then referenced in the context variables.

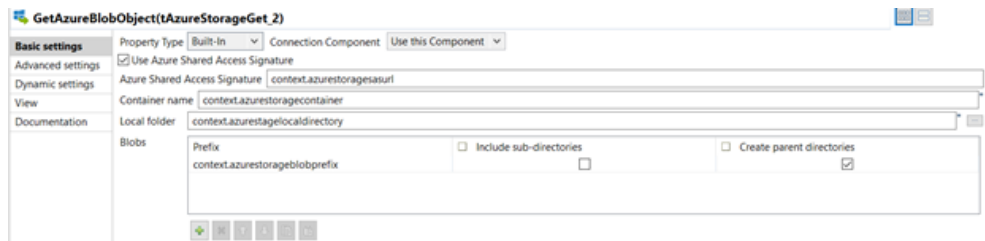
B. Configure each Talend component

• tAzureStorageGet

Select the 'Use Azure Shared Access Signature' checkbox.

Specify the context variables for Azure Shared Access Signature, Container name, Local folder and Blobs prefix.

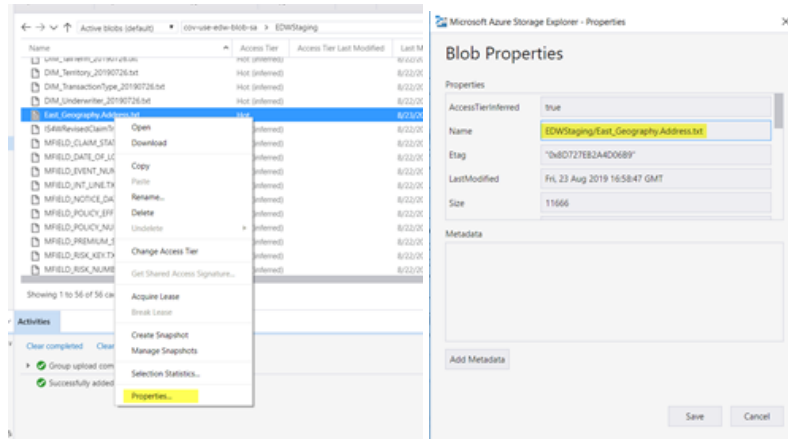
Select the 'Create Parent directories' checkbox.



Additional information on Parameters:

Prefix: this allows you to filter the blobs which have the specified prefix in their names in the given container. For the flat files, you need to specify the file name, including the full path. You can get this information from Azure Storage Explorer by right clicking a file and selecting properties, and then copying the 'Name' field. A blob name contains the virtual hierarchy of the blob itself. This hierarchy is a virtual path to that blob and is relative to the container where that blob is stored.

For example, in a container named cov-use-edw-blob-sa, the name of a flat file blob might be EDW_Staging/East_Geography.Address.txt.



Include sub-directories: select this check box to retrieve all of the sub-folders and the blobs in those folders beneath the designated directory level in the Blob prefix column. If you leave this check box clear, tAzureStorageGet returns only the blobs directly beneath that directory level. We do not need to check this since we're fetching one specific file at a time.

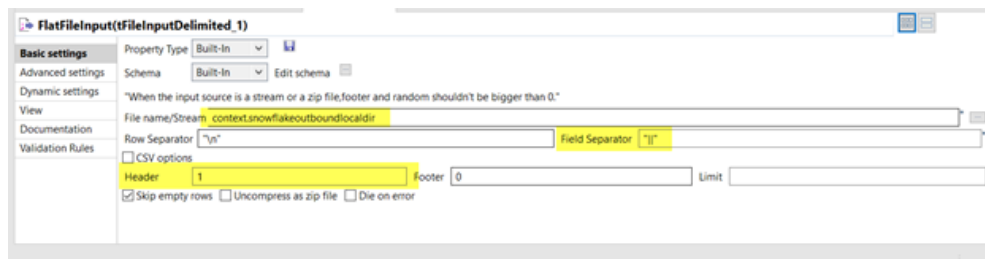
Create parent directories: select this check box to replicate the virtual directory of the retrieved blobs in the local folder. Note that if you leave this check box clear, there must be the same directory in the local folder as the retrieved blobs have in the container; otherwise, those blobs cannot be retrieved. Refer FAQ section below for associated errors.

• tFileInputDelimited

Specify the context variable for file name in the 'File name/Stream' field. If your data contains date columns, ensure the format in 'Edit schema' window matches the actual date formats in your data.

Specify the Field separator and Header values.

Open Edit schema, and include StagingExecutionID and ExtractionExecutionID, and specify the corresponding type and context variables.



Schema of FlatFileInput

Column	K.	Type	N.	Date Pattern (C...	Length	Precision	Default	Comment
ADDRESSKEY		String	<input checked="" type="checkbox"/>					
LOCATIONTYPE		String	<input checked="" type="checkbox"/>					
LONGITUDE		String	<input checked="" type="checkbox"/>					
METROPOLITANAREA		String	<input checked="" type="checkbox"/>					
POSTALCODE		String	<input checked="" type="checkbox"/>					
POSTALCODEEXT		String	<input checked="" type="checkbox"/>					
SOURCESYSTEMCODE		String	<input checked="" type="checkbox"/>					
SOURCESYSTEMDATECREATED		Date	<input checked="" type="checkbox"/>	"MM/dd/yyyy"				
SOURCESYSTEMDATEMODIFIED		Date	<input checked="" type="checkbox"/>	"MM/dd/yyyy"				
STATEABBREVIATION		String	<input checked="" type="checkbox"/>					
STATENAME		String	<input checked="" type="checkbox"/>					
ADDRESSLINE1		String	<input checked="" type="checkbox"/>					
STATENUMERICCODE		BigDecimal	<input checked="" type="checkbox"/>					
STATISTICALAREA		String	<input checked="" type="checkbox"/>					
EFFECTIVEDATE		Date	<input checked="" type="checkbox"/>	"MM/dd/yyyy"				
EXPIRATIONDATE		Date	<input checked="" type="checkbox"/>	"MM/dd/yyyy"				
ADDRESSLINE2		String	<input checked="" type="checkbox"/>					
ADDRESSLINE3		String	<input checked="" type="checkbox"/>					
CENSUSBLOCK		String	<input checked="" type="checkbox"/>					
CITY		String	<input checked="" type="checkbox"/>					
COUNTRY		String	<input checked="" type="checkbox"/>					
COUNTY		String	<input checked="" type="checkbox"/>					
LATITUDE		String	<input checked="" type="checkbox"/>					
STAGINGEXECUTIONID		int	<input checked="" type="checkbox"/>					
EXTRACTIONEXECUTIONID		int	<input checked="" type="checkbox"/>					

context.stagingexecutionid
context.extractionexecutionid

OK Cancel

• tSnowflakeOutput

Specify the Connection details, and select the correct table name.

Select 'TRUNCATE' under Table Action.

Job Contexts(Test_Azure_Blob_to_Snowflake_DEV) Component Run (Job Test_Azure_Blob_to_Snowflake_DEV) Test Cases Cloud Artifact

Snowflake_IS4EAST_GEOGRAPHY_ADDRESS(tDBOutput_3)(Snowflake)

Basic settings Database: Snowflake Apply

Advanced settings Property Type: Repository snowflake:Conn_Snowflake_Staging Connection Component: Use this Component

Dynamic settings Account: context.Conn_Snowflake_Staging_IS4EAST_account

View Snowflake Region: context.Conn_Snowflake_Staging_IS4EAST_region

Documentation User Id: context.Conn_Snowflake_Staging_IS4EAST_userPassword_userid

Validation Rules Password: context.Conn_Snowflake_Staging_IS4EAST_userPassword_password

Warehouse: context.Conn_Snowflake_Staging_IS4EAST_warehouse

Schema: context.Conn_Snowflake_Staging_IS4EAST_schemaName

Database: context.Conn_Snowflake_Staging_IS4EAST_db

Table: "GEOGRAPHY_ADDRESS"

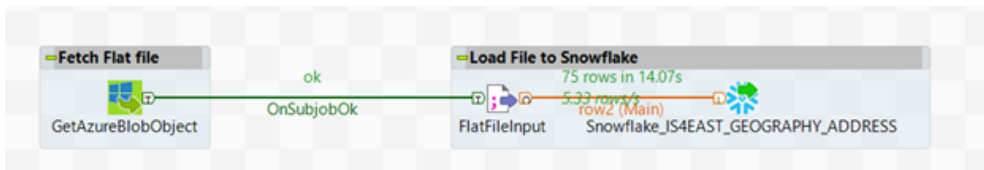
Schema: Built-In Edit schema Sync columns

Table Action: TRUNCATE

Output Action: INSERT

C. Complete your Talend Job

Make necessary connections as shown below to complete the configuration of your Talend job.

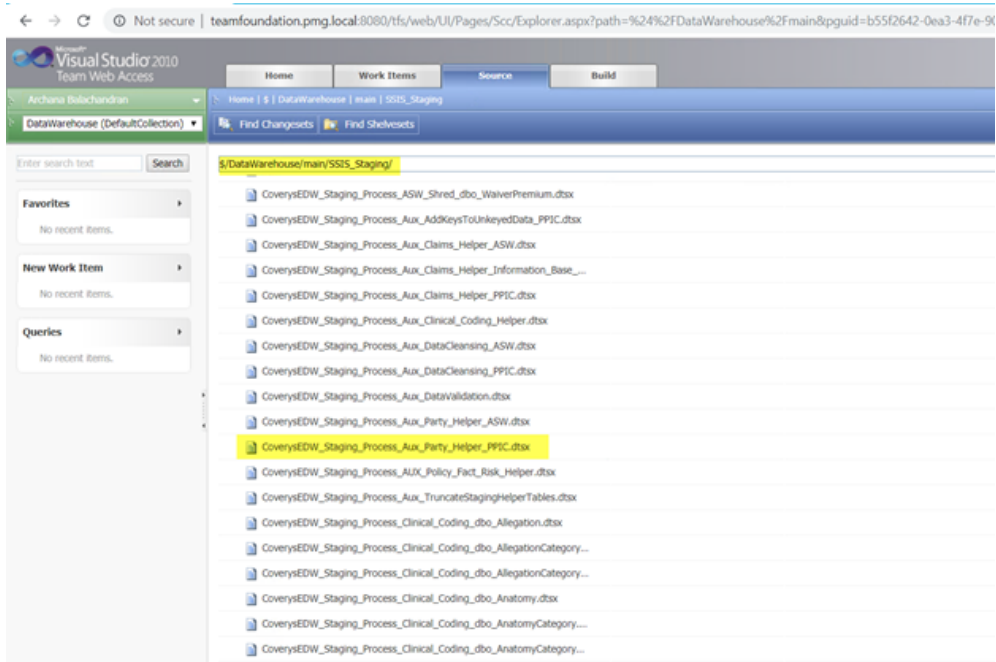


SSIS Package to SnowSQL Conversion

The objective of this task is to convert the latest SSIS package for the package assigned, and convert them into a series of SnowSQL scripts. For each step within the SSIS Package, it is necessary to convert the SQL from TransactSQL to SnowSQL. We are selecting the package for Party Helper ASW as an example in this document. Package name = CoverysEDW_Staging_Process_Aux_Party_Helper_ASW.dtsx

1. Download and View Latest SSIS Package in Visual Studio

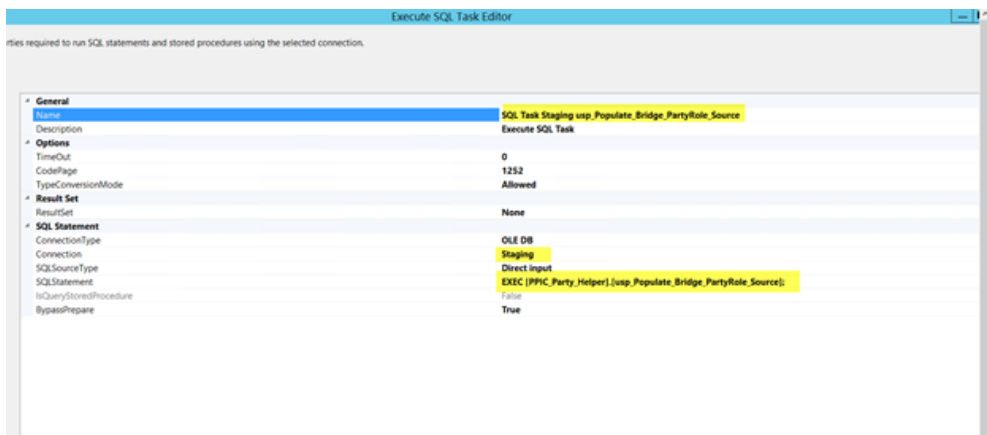
Download the latest SSIS package from SSIS_Staging folder from Visual Studio Team Web Access site.



Open the file in Visual Studio, and view all the tasks under 'Work'. For each task, note the stored procedure being executed.

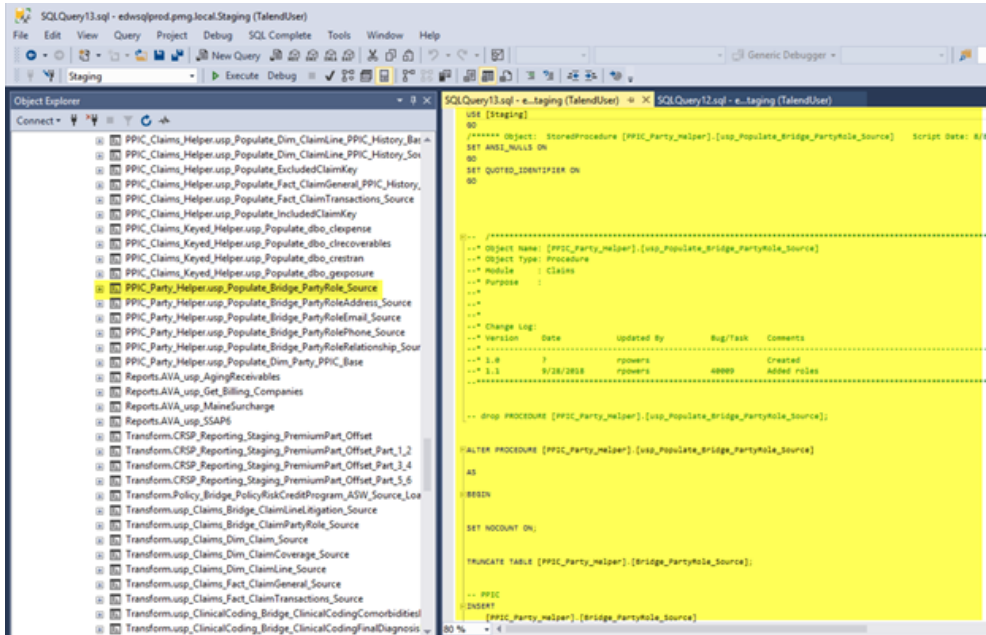


For example, the first task 'SQL Task Staging usp_Populate_Bridge_PartyRole_Source' connects to 'Staging' database and executes stored procedure [PPIC_Party_Helper].[usp_Populate_Bridge_PartyRole_Source];.



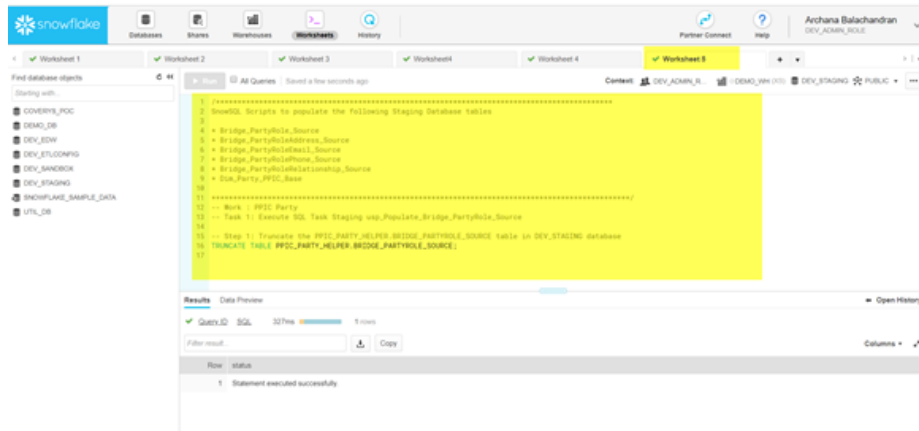
2. View SQL Code for Stored Procedure in SQL Server Management Studio

Open SQL Server Management Studio and navigate to the required Stored Procedure and view the code.



3. Convert the Transact SQL Code to SnowSQL in Snowflake GUI

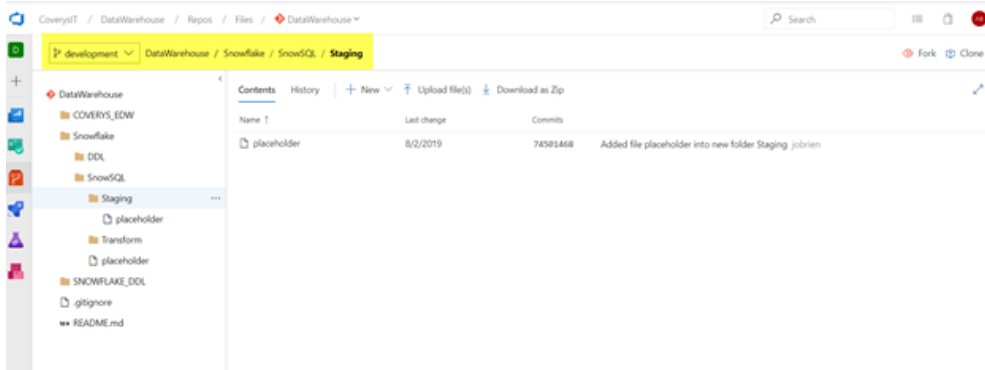
Development/Design can take place in the Snowflake GUI providing the required source staging tables have been populated. Ensure that the code is executed in Snowflake console before saving into a .sql file.



All tasks within 'Work' can be combined and saved into one .sql file, if there are no other dependencies for their execution.

4. Update SnowSQL scripts in Git Repository

Please make sure your scripts are periodically checked into Git, in the location specified in the screenshot below. The naming convention for the file is the same as SSIS package naming convention, followed by a .Sql extension. For example, 'CoverysEDW_Staging_Process_Aux_Party_Helper_ASW.sql' will be the name of the script we walked through in this document.



For information on GitHub Desktop setup, please refer the following link:

https://ecoverys.sharepoint.com/:w:/r/sites/extEDWCloudMigration/_layouts/15/Doc.aspx?sourcedoc={02160515-27E9-4B61-BE71-6D897005A4A8}&file=Talend_Install_and_Configuration_Git_Hub_Overview.docx&action=default&mobileredirect=true

NOTES AND HELPFUL RESOURCES:

Notes:

- Write all the SnowSQL scripts for tasks specified in package in one file.
- Encapsulating table names or column names in quotes make them case sensitive. Avoid typing objects in quotes unless it is required.
- It is best practice to specify the columns you are inserting into in your 'INSERT INTO table_name' statement.
- You can alias a column using the 'AS' keyword. The 'AS' keyword is also used to assign values to columns.

Snowflake Community Documentation:

Topic	Link
SnowSQL Commands Glossary	https://docs.snowflake.net/manuals/sql-reference/sql-all.html
Pivot Function in SnowSQL	https://docs.snowflake.net/manuals/sql-reference/constructs/pivot.html
Unpivot	https://docs.snowflake.net/manuals/sql-reference/constructs/unpivot.html
SnowSQL Query Syntax	https://docs.snowflake.net/manuals/sql-reference/constructs.html

Common SQL to SnowSQL Conversions

SQL	SnowSQL
[RowsCurrent] = CAST('Y' AS [nchar] (1))	CAST('Y' AS VARCHAR) AS ROWISCURRENT
[RowEndDate] = CAST('12/31/9999' AS [datetime])	CAST('12/31/9999' AS datetime) AS ROWENDDATE
ColumnName='Value'	'Value' AS ColumnName
ISNULL(expr1,expr2)	IFNULL(expr1,expr2)

For a detailed example on conversion of a script, refer:

- [https://ecoverys.sharepoint.com/sites/extEDWCloudMigration/Documents/Sprint%203_%20SSIS%20Package%20to%20SnowSQL%20Conversion%20\(Helper%20Tables\).docx?d=w12c2299dfcb64864b9b623feb1a1f0af](https://ecoverys.sharepoint.com/sites/extEDWCloudMigration/Documents/Sprint%203_%20SSIS%20Package%20to%20SnowSQL%20Conversion%20(Helper%20Tables).docx?d=w12c2299dfcb64864b9b623feb1a1f0af)
- https://ecoverys.sharepoint.com/sites/extEDWCloudMigration/Documents/Sprint3_SnowSQLScript_PPIC_Party_Helper_usp_Bridge_Party_Role_Source.sql