This document outlines the steps involved in generating or creating SSIS packages using an automation process; which uses configuration tables, xml code and BCP commands to create SSIS package in bulk.

*Note: The automation process outlined below for SSIS package creation works with package using ‘Execute SQL Task’ only.*

# Steps

1. Create a sample SSIS package template
2. Insert and Update configuration tables
3. Generate BCP command
4. Execute BCP command from command prompt

# Create SSIS Project & Package

As part the automation, we are required to create a sample package with all required ETL components for the data load process.

As part of M360 v4, any SSIS package used for data load in M360 data warehouse should use the SSIS wrapper. The SSIS wrapper built for v4, checks for license validation and logs the results in tables under [M360\_V4\_ADMIN\_DATABASE] database. Along with license validation result, the wrapper also logs high level issues due to which the packaged execution failed. The table used for logging the above mentioned detail is named as M360\_JOB\_METADATA

As a result, we are required to use SSIS wrapper for automation process as well.

1. Create a new SSIS project based on the sample project available in Surround.

Surround Location:

This project has 4 project parameters used by the SSIS wrapper.

This project also has a sample package ‘ETL\_ODS\_LOAD\_TEMPLATE.dtsx’ for ODS data load process. This package has variables defined used by wrapper and for ODS data load process.

*Except for BatchAsOfDate & ListOfDates variables, the variables defined in the package are used by the SSIS wrapper.*

*BatchAsOfDate & ListOfDates variables were added for ODS data load process.*

1. Make sure to rename the new project created based on the database name for that core system.

Example: If we creating a new project to loading data into ULTRADATA\_ODS database, then the project should be named as Ultradata\_ODS; for datasafe it would be DATASAFE\_ODS etc.

# Configuration Database & Tables

For SSIS package automation process, we would populate 2 database tables with all required code for the package. The configuration tables used for the automation process would be defined under database **[PACKAGE\_BUILD]**. The naming convention for the configuration tables would be –

* <core system name>\_ODS\_PACKAGE\_CODE
* <core system name>\_ODS\_PACKAGE\_XML

**Surround Location:**

* Database Create Script

//M360 v4.0.0 - Development//M360 Standard/Connector Core – UltraData/Scripts/SQL Server/SQL Database/PACKAGE\_BUILD/

File name: PACKAGE\_BUILD.sql

* Table Create Script

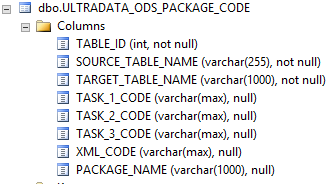
//M360 v4.0.0 - Development//M360 Standard/Connector Core – UltraData/Scripts/SQL Server/SQL Objects/PACKAGE\_BUILD/

File name: Create ODS Package Automation Configuration Tables.sql

*Note: The table names mentioned below were created for ULTRADATA\_ODS package automation. When working on a new or different core ODS database, new tables would be created with similar structure and the script to go the respective folder in Surround.*

1. [dbo].[ULTRADATA\_ODS\_PACKAGE\_CODE]

This table hold the actual code for each of the SSIS package which would generated as part of the automation.



* TABLE\_ID – this is an identity column
* SOURCE\_TABLE\_NAME & TARGET\_TABLE\_NAME – this would hold the ODS table name
* TASK\_1\_CODE – the column should be populated with the SQL code for the first EXECUTE SQL task in the template SSIS package
* TASK\_2\_CODE – the column should be populated with the SQL code for the second EXECUTE SQL task in the template SSIS package
* TASK\_3\_CODE – the column should be populated with the SQL code for the third EXECUTE SQL task in the template SSIS package
* XML\_CODE – the column would hold the complete XML code for the package to be created
* PACKAGE\_NAME – the column should be populated with the ODS package name for that table

1. [dbo].[ULTRADATA\_ODS\_PACKAGE\_XML]

This table has the xml code of the SSIS template package described above.

Use the below mentioned steps to populate the above mentioned tables & columns –

* ULTRADATA\_ODS\_PACKAGE\_XML

INSERT INTO PACKAGE\_BUILD.[dbo].[ULTRADATA\_ODS\_PACKAGE\_XML](XMLData)

SELECT CONVERT(VARCHAR(MAX), BulkColumn) AS BulkColumn

FROM OPENROWSET(BULK 'C:\SG\ODS\_Template.dtsx', SINGLE\_BLOB) AS x;

*Note: C:\SG\ - this is a temp folder which I used for placing the template package, used in above SQL. To create this template package – right click on sample package “ETL\_ODS\_LOAD\_TEMPLATE.dtsx” and select ‘View Code’. Copy the xml code and paste it on to a notepad. Save the notepad file named as ODS\_Template.dtsx;* ***make sure that while saving this notepad as dtsx file we select ‘All Files’ option under “Save as type:****’.*

* ULTRADATA\_ODS\_PACKAGE\_CODE
  + SQL code for populating SOURCE\_TABLE\_NAME, TARGET\_TABLE\_NAME, TASK\_1\_CODE, TASK\_3\_CODE & PACKAGE\_NAME

USE ULTRADATA\_ODS

INSERT INTO PACKAGE\_BUILD.[dbo].[ULTRADATA\_ODS\_PACKAGE\_CODE]

([SOURCE\_TABLE\_NAME]

,[TARGET\_TABLE\_NAME]

,[TASK\_1\_CODE]

,[TASK\_3\_CODE]

,[PACKAGE\_NAME]

)

SELECT TABLE\_NAME,

TABLE\_NAME,

'SELECT DISTINCT BATCH\_AS\_OF\_DATE FROM ' + TABLE\_NAME + ' WHERE BATCH\_AS\_OF\_DATE IS NOT NULL ORDER BY BATCH\_AS\_OF\_DATE',

'DELETE FROM STAGE..' + TABLE\_NAME + ' WHERE BATCH\_AS\_OF\_DATE = ?',

'ETL\_LOAD\_ULTRADATA\_ODS\_' + TABLE\_NAME

FROM INFORMATION\_SCHEMA.TABLES

Note: The above mentioned SQL has where clause as a result it will consider all tables in ULTRADATA\_ODS database. Make sure you use the right database and filter condition if any before executing the above SQL.

* + SQL to populate TASK\_2\_CODE

The ODS table load process uses merge logic. Hence, the MERGE SQL for each ODS table should go in this column.

We have a store procedure which can be used to generate MERGE SQL for a given ODS table. The stored procedure takes ODS table name as an input and assumes that the table name on STAGE database is same; which will be source for ODS table. Stored Procedure is available in Surround –

//M360 v4.0.0 - Development//M360 Standard/Connector Core – UltraData/Scripts/SQL Server/SQL Objects/PACKAGE\_BUILD/

File Name: GenerateMergeStatement.sql

* + Update XML\_CODE
    - Update XML\_CODE from ULTRADATA\_ODS\_PACKAGE\_XML table

UPDATE PACKAGE\_BUILD..[ULTRADATA\_ODS\_PACKAGE\_CODE]

SET XML\_CODE = X.XMLData

FROM PACKAGE\_BUILD..ULTRADATA\_ODS\_PACKAGE\_XML X

* Update XML\_CODE on ULTRADATA\_ODS\_PACKAGE\_CODE with TASK\_1\_CODE

UPDATE PACKAGE\_BUILD.[dbo].[ULTRADATA\_ODS\_PACKAGE\_CODE]

SET XML\_CODE = P.UPDATED\_XML

FROM PACKAGE\_BUILD..[ULTRADATA\_ODS\_PACKAGE\_CODE] PC

INNER JOIN (SELECT TABLE\_ID, I.XML\_CODE, REPLACE(I.XML\_CODE, 'SELECT DISTINCT CAST(GETDATE() AS DATE) AS DATE\_PART',I.TASK\_1\_CODE) UPDATED\_XML

FROM PACKAGE\_BUILD..[ULTRADATA\_ODS\_PACKAGE\_CODE] I

) P ON PC.TABLE\_ID = P.TABLE\_ID

* Update XML\_CODE on ULTRADATA\_ODS\_PACKAGE\_CODE with TASK\_2\_CODE

UPDATE PACKAGE\_BUILD.[dbo].[ULTRADATA\_ODS\_PACKAGE\_CODE]

SET XML\_CODE = P.UPDATED\_XML

FROM PACKAGE\_BUILD..[ULTRADATA\_ODS\_PACKAGE\_CODE] PC

INNER JOIN (SELECT TABLE\_ID, I.XML\_CODE, REPLACE(I.XML\_CODE, 'SELECT ?',I.TASK\_2\_CODE) UPDATED\_XML

FROM PACKAGE\_BUILD..[ULTRADATA\_ODS\_PACKAGE\_CODE] I

) P ON PC.TABLE\_ID = P.TABLE\_ID

* + - Update XML\_CODE on ULTRADATA\_ODS\_PACKAGE\_CODE with TASK\_3\_CODE

UPDATE PACKAGE\_BUILD.[dbo].[ULTRADATA\_ODS\_PACKAGE\_CODE]

SET XML\_CODE = P.UPDATED\_XML

FROM PACKAGE\_BUILD..[ULTRADATA\_ODS\_PACKAGE\_CODE] PC

INNER JOIN (SELECT TABLE\_ID, I.XML\_CODE, REPLACE(I.XML\_CODE, 'SELECT 3, ?',I.TASK\_3\_CODE) UPDATED\_XML

FROM PACKAGE\_BUILD..[ULTRADATA\_ODS\_PACKAGE\_CODE] I

) P ON PC.TABLE\_ID = P.TABLE\_ID

* + - Update XML\_CODE on ULTRADATA\_ODS\_PACKAGE\_CODE to change default package name to correct name based on the table name

UPDATE PACKAGE\_BUILD.[dbo].[ULTRADATA\_ODS\_PACKAGE\_CODE]

SET XML\_CODE = P.UPDATED\_XML

FROM PACKAGE\_BUILD..[ULTRADATA\_ODS\_PACKAGE\_CODE] PC

INNER JOIN (SELECT TABLE\_ID, I.XML\_CODE, REPLACE(I.XML\_CODE, 'ETL\_ODS\_LOAD\_TEMPLATE',I.PACKAGE\_NAME) UPDATED\_XML

FROM PACKAGE\_BUILD..[ULTRADATA\_ODS\_PACKAGE\_CODE] I

) P ON PC.TABLE\_ID = P.TABLE\_ID

# Generate BCP Script

SQL given below will generate the BCP command which needs to be executed from the command prompt. We can create the new packages in a different folder other than the SSIS project folder and once we have the packages created, we can then use ‘Add Existing Package’ option to add the new packages created to the SSIS project created in step 1 above.

SELECT 'bcp "SELECT [XML\_CODE] FROM PACKAGE\_BUILD.[dbo].[ULTRADATA\_ODS\_PACKAGE\_CODE] WHERE TARGET\_TABLE\_NAME = ''' + T.TABLE\_NAME

+ '''" queryout C:\SG\' + T.PACKAGE\_NAME + '.dtsx -c -T -S OADEVSERVER2012\M360\_OA\_00000 '

FROM (SELECT TARGET\_TABLE\_NAME TABLE\_NAME , PACKAGE\_NAME FROM PACKAGE\_BUILD.[dbo].[ULTRADATA\_ODS\_PACKAGE\_CODE]) T