APS to Azure SQL Data Warehouse Migration



Migration - Data Migration with BCP and SSIS

Prepared for

17-Jan-20

Version 1.0 Draft

Contributors

**Arshad Ali,** Solution Architect

**Kalyan Yella, Solution Architect**

**Vishal Singh**, Consultant

**Basawashree Vasmate**, Consultant

Table of Contents

[1 Introduction 2](#_Toc486423531)

[2 Data Migration Flow 3](#_Toc486423532)

[2.1 Step 1: 4](#_Toc486423533)

[2.2 Step 2: 5](#_Toc486423534)

[3 Runtime Configuration Parameters 7](#_Toc486423535)

[4 Logging and Recovering from Failure 8](#_Toc486423536)

1. Introduction

In certain cases when you cannot use Polybase to export the data out from APS, you can use BCP method. This document talks about the tool which migrates the data from APS database to the file system using BCP and SSIS (for parallel exports). The tool generates a folder for each table in the APS, containing a gzip file having the data of the respective table.

The data export process has these two steps:

1. **Execute the sql script on the APS database**

This step creates a table called ‘PartitionInformation’ in the database and populates it with all the necessary commands required for data migration.

1. **Run the SSIS Package (Optional)**

In the second step, SSIS package will fetch the commands from table created in step 1 and executes in multiple parallel threads. This step is an optional as the commands from the table, created in step 1, can also be manually executed using command prompt.

1. Data Migration Flow

Based on the type of the source table (whether partitioned or not), the tool generates output files. For a non-partitioned table, it generates single file and export all the data from the table in that single file whereas in case of partitioned table, it generates one file for each partition of the table and export data from that corresponding partition of the table. It also uses 7zip utility (that’s a prerequisite if you want to compress the output files or else comment out that component in SSIS package) to compress the output files in GZIP archival format, which is supported when using Polybase to import data into SQL Data Warehouse. Use of compression, obviously helps in optimizing data movement and recommended to use it.

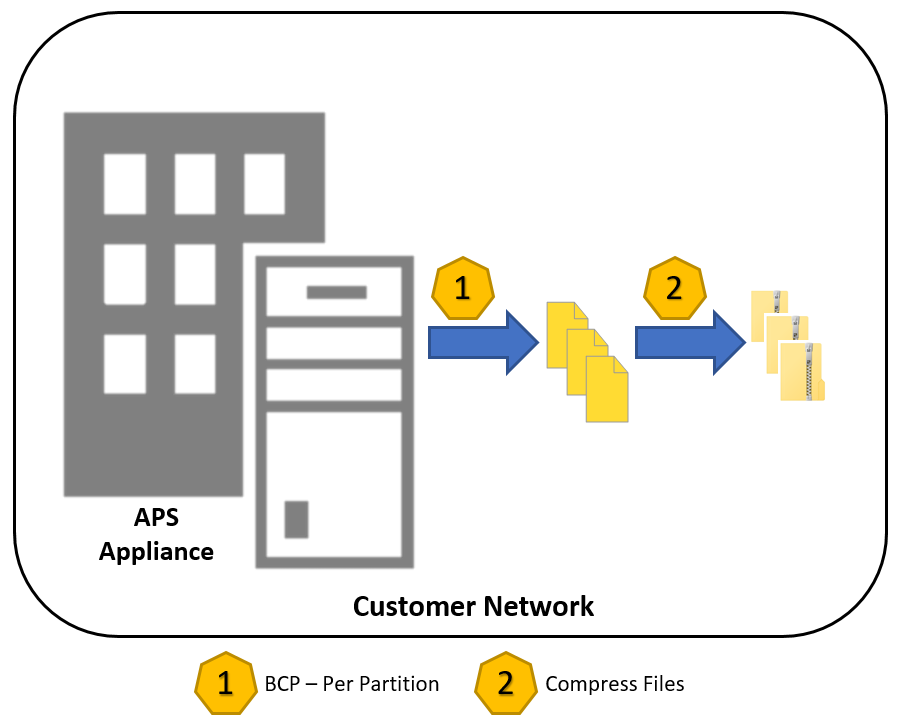


Figure 1 - Data Export with BCP and SSIS

* 1. Step 1:

In this step, a table is created and populated as per the below procedure.

**For Non-Partitioned Tables:**

There will be one row for each non-partitioned table, having below details

***DBName*** - Database name

***SchemaName*** - schema name

***Tablename*** – Table name

***PartitionNumber*** – 1 (default for non-partitioned table)

***DirCreateCommand*** – it has command to create a Folder with same name as the Table name at the specified location in the file system.

***BCPCommand*** – it has Command to export table data into a file with the same name as Table name.

***GZIPCommand*** – it has command to compress the file generated by executing the command in above step

***DelFileCommand*** –It has command to delete the decompressed file.

**For Partitioned Tables:**

There will be one row for each partition of the table, having below details

***DBName*** - Database name

***SchemaName*** - schema name

***Tablename*** – Table name

***Type*** – Type of the partition (Left/Right)

***PartitionNumber*** – Partition number

***UpperBoundaryValue*** – UpperBoundaryValue for the partition.

***LowerBoundaryValue –*** LowerBoundaryValue for the partition.

***DirCreateCommand*** – it has command to create a Folder with same name as the Table name at the specified location in the file system.

***BCPCommand*** – it has Command to export that particular partition data into a file with the same name as Table name suffixed by ‘\_<UpperBounadryValue>’.

This command uses UpperBoundaryValue, LowerBoundaryValue and Type columns to filter the table data respective to the defined partition.

***GZIPCommand*** – it has command to compress the file generated by executing the command in above step

***DelFileCommand*** –It has command to delete the decompressed file.

* 1. Step 2:

In this step, you need to execute the SSIS package. Structure of the package is as below

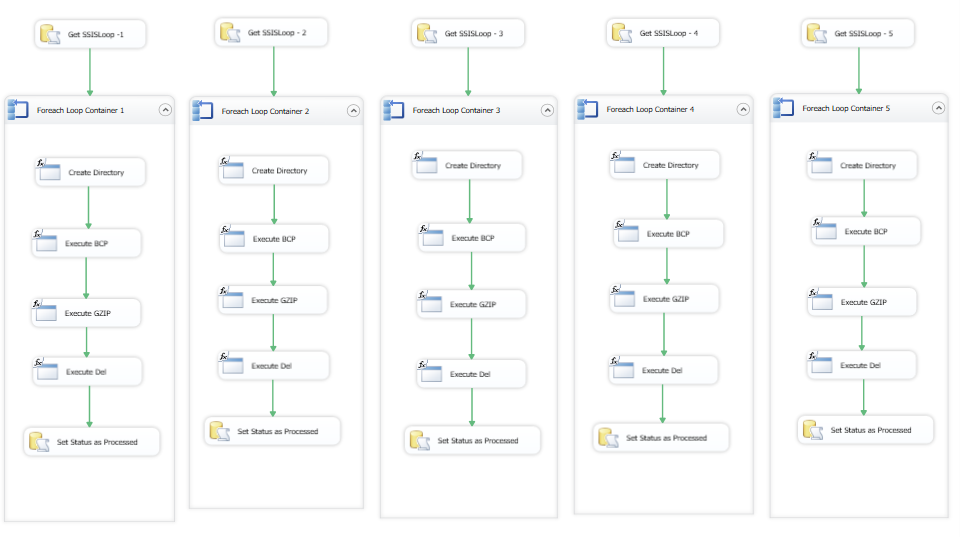


Table in the Step 1 will also have a column ‘SSISLoop’ which is populated with sequential number for each row, starting from 1 up to a configurable number (5 or 10) equal to maximum concurrency level in SSIS.

There are 2 packages to facilitate the parallelism, one will process 5 rows or 5 exports in parallel, and another will process 10 rows or 10 exports in parallel at a time.

As depicted in the picture above,

Loop1 will pick all the rows from the table where SSISLoop column value = 1

Loop2 will pick all the rows from the table where SSISLoop column value = 2 and so on

And processed the create directory command, BCP command, GZIP command and delete file command sequentially from each row.

1. Runtime Configuration Parameters

These are some of the important configuration parameters which need to be defined before execution of the step 1 script, for the tool to function correctly:

* **@APSServer:** IP of the APS Appliance
* **@Username:** User name which will be used to connect APS
* **@Password:** Password for above user name
* **@ZIPExePathName:** Zip Exe full Path name
* **@NoOfConcurrentSSISLoop:** 5 or 10 (to populate SSISLoop column and accordingly should choose SSIS package to be executed)
* **@FilePath:** File Path in which all the folders to be created for each table in the APS database.

1. Logging and Recovering from Failure

The tool logs execution information in the same table created in Step 1. It sets a flag ‘IsProcessed’ in the table ‘PartitionInformation’ to 1 on successful export of the data to file. By default, the value is 0. This means if the export process fails during execution, the subsequent execution of the package will pick up the export from the point where it failed last time – (skipping all those exports which completed successfully in last attempts).