# 

APS to Azure Synapse Analytics Migration Guide

* With GitHub ([link](https://github.com/microsoft/AzureSynapseScriptsAndAccelerators/tree/main/Migration/APS)) Open Source Tools

APS to Azure Synapse Analytics Migration Guide using Open Source Tools Published in GitHub ([link](https://github.com/microsoft/AzureDWScriptsandUtilities/tree/master/APS%20to%20SQL%20DW%20Migration%20-%20Schema%20and%20Data%20Migration%20with%20PolyBase/6_DeployScriptsToSqldw))

Prepared for APS to Azure Synapse Analytics Migration Practitioners

**Prepared by**

**Gaiye ‘Gail’ Zhou**

Solution Architect

[gazho@microsoft.com](mailto:gazho@microsoft.com)

Azure Synapse Analytics Customer Success Engineering

**Version**: V1.0

Last Update: March 26, 2020

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
| March 2021 | Gaiye ‘Gail’ Zhou | 1.3 | Initial version |
| June 2021 | Aruna Dadi Andrey Mirskiy | 1.4 | Updates to reflect latest PowerShell toolkit version |

Table of contents

[Change Record 2](#_Toc74671083)

[Migration Step by Step Guide 1](#_Toc74671084)

[Environment and Workstation Preparations 1](#_Toc74671085)

[1.1. Prepare APS Environments 1](#_Toc74671086)

[1.2. Create Azure Subscription and Resources 1](#_Toc74671087)

[1.3. Setup Workstation(s) or Laptop(s) 3](#_Toc74671088)

[1.4. Prepare APS (T-SQL Scripts) 4](#_Toc74671089)

[1.5. Prepare Azure Synapse Analytics (T-SQL Scripts) 4](#_Toc74671090)

[1.6. Prepare Schema Mapping Matrix 4](#_Toc74671091)

[Download and Setup the Open Source Migration Tools 5](#_Toc74671092)

[2.1. Download Migration Tools 5](#_Toc74671093)

[2.2. Setup PowerShell Scripts Permissions 6](#_Toc74671094)

[Implementation Steps and Output 8](#_Toc74671095)

[3.1. Step 1: Obtain APS DDLs and DMLs 8](#_Toc74671096)

[3.2. Step 2: Format APS DDLs and DMLs 8](#_Toc74671097)

[3.3. Step 3: Translate APS DDLs and DMLs 8](#_Toc74671098)

[3.4. Step 4: Create Export and Import T-SQL Scripts 8](#_Toc74671099)

[3.5. Step 5: Create T-SQL Scripts for Azure Synapse External Tables 8](#_Toc74671100)

[3.6. Step 6: Deployments 9](#_Toc74671101)

[3.7. Spot check Table and stored procedures and resolve issues 9](#_Toc74671102)

[3.8. POC Results Verification Tests 9](#_Toc74671103)

[Migration Exceptions Handling 10](#_Toc74671104)

[4.1. Additional Code Conversion 10](#_Toc74671105)

[4.2. Step 6: Deployments 10](#_Toc74671106)

[Appendix T-SQL Scripts (Samples) 10](#_Toc74671107)

Migration Step by Step Guide

# Environment and Workstation Preparations

Note: Please extract the zip file named T-SQL-Scripts-Samples.zip in (Appendix T-SQL Scripts (Samples)).

## Prepare APS Environments

* + **Set up Polybase in APS** – Please refer to Setup\_APS\_Polybase\_Sample.sql for details.
  + **Check APS Collation using T-SQL**

SELECT CONVERT (varchar, SERVERPROPERTY('collation')) AS 'Server Collation';

Sample Return Result: Latin1\_General\_100\_CI\_AS\_KS\_WS

## Create Azure Subscription and Resources

* + **Azure Subscription**: Name here
  + **Azure Resource Region**: North Europe
  + **Azure Storage (Blob Storage) Name:**  Blob Storage Account Name
    - Storage Settings for Polybase
      1. **Performance**: Standard
      2. **Security transfer required**: Enabled
      3. **Access tier** (default): Hot
      4. **Replication**: Locally redundant storage (LRS)
    - Connectivity: Public endpoint (all networks)
    - Security
      1. Security transfer required: Enabled
      2. Blob soft deletion: Disabled
      3. Data Lake Storage Gen2 (Hierarchical namespace): Disabled
    - Create containers with below suggested names:
      1. pocmigrations: Use this container for APS->ASA migration POC.
      2. pocdelta: Use this container for data ingestion process (Sources->ASA) POC.
      3. migrations: Use this container for production migrations.
  + Azure Storage General Purpose V2 (Blob Storage)

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

* + **Azure Synapse Analytics:** Server name and db name
    - Create Azure Synapse Analytics (formerly Azure SQL DW). Collation Choice examples:
      1. Use Azure Synapse default collation which is “SQL\_Latin1\_General\_CP1\_CI\_AS”. Then all the APS creates table statements need to replace original APS collation with blank. For example, Replace “Latin1\_General\_100\_CI\_AS\_KS\_WS” with blank string “ “. If you don’t replace them with empty string, you may encounter problems. Please do this.
      2. Create Azure Synapse Analytics with default collation unless otherwise specified by the customer. This applies to most of the situations.
      3. Create Azure Synapse Analytics (formerly Azure SQL DW) with the same collation as current APS collation if that is appropriate / required.
    - In Azure Synapse Analytics, the Collation cannot be changed after data warehouse creation. The default collation is SQL\_Latin1\_General\_CP1\_CI\_AS. For more details, please read this online [documentation page](https://docs.microsoft.com/en-us/sql/t-sql/statements/collations?view=sql-server-ver15).
    - In ASA, Check Azure SQ DW Collation by the following T-SQL

SELECT CONVERT (varchar, SERVERPROPERTY('collation')) AS 'Server Collation';

Sample Return Result:

ADW: SQL\_Latin1\_General\_CP1\_CI\_AS

APS: Latin1\_General\_100\_CI\_AS\_KS\_WS

* + - Check Azure SQ DW DB Collation by the following T-SQL

SELECT DATABASEPROPERTYEX(DB\_NAME(), 'Collation') AS Collation;

Sample Return Result: SQL\_Latin1\_General\_CP1\_CI\_AS

* + - List all supported collations in Azure SQ DW by the following T-SQL (connect to master)

SELECT \* FROM sys.fn\_helpcollations();

* + **Azure Active Directory & Security Setup** 
    - Onboarding project team members to access customer network (Laptop issued by customer in some cases)
    - Set up access for project team members to access Azure Resources (Customer Azure Security Admin performs this after all Azure Resources have been created).

## Setup Workstation(s) or Laptop(s)

* + **Network connectivity**: Connectivity to Internet, APS, and Customer Azure Resources.
  + **Customer team communications**: Connectivity to Customerinternal communications: Email, team sites, etc.
  + **Data Tools Setup** 
    1. Visual Studio 2017 and SSDT to run APS T-SQL Scripts (This is required for APS access)
    2. SSMS: Latest Version to run ASA T-SQL Scripts (optional but recommended)
    3. Visual Studio Code or other user preferred IDEs to run PowerShell Scripts and Python Programs

## Prepare APS (T-SQL Scripts)

* + **T-SQL Scripts**: Setup\_APS\_Polybase.sql (Run by PFE or APS Admin)
  + **T-SQL Scripts:** Setup\_APS\_Migration\_User.sql (Run by PFE or APS Admin)
  + **T-SQL Scripts:** Setup\_APS\_Export.sql (Run by Migration User)

Sample T-SQL Scripts are provided in Section 4, “Migration Exceptions Handling“. For APS roles and user permission, please refer to APS Document (click [here](https://docs.microsoft.com/en-us/sql/analytics-platform-system/pdw-permissions?view=aps-pdw-2016-au7#fixed-database-roles)).

## Prepare Azure Synapse Analytics (T-SQL Scripts)

* + **T-SQL Scripts:** Setup\_ASA\_Migration\_User.sql (Run by PFE or ASA Admin)
  + **T-SQL Scripts:** Setup\_ASA\_Import.sql (Run by Migration User)

Sample T-SQL Scripts are provided in Section 4, “Migration Exceptions Handling“.

## Prepare Schema Mapping Matrix

* + **Prepare Schema Mapping File:** schemas.csv
  + **Below is a sample schema mapping file** (contents of schemas.csv)**:**

|  |  |  |
| --- | --- | --- |
| ApsDbName | ApsSchema | SQLDWSchema |
| adventureworks\_stage | dbo | stage\_dbo |
| adventureworks\_edw | dbo | edw\_dbo |
| adventureworks\_edw | hr | edw\_hr |
| adventureworks\_datamarts | hr | dm\_hr |
| adventureworks\_datamarts | finance | dm\_finance |

# Download and Setup the Open Source Migration Tools

## Download Migration Tools

The [Github](https://github.com/microsoft/AzureSynapseScriptsAndAccelerators/tree/main/Migration/APS) contains overview of the migration process and very detailed step by step instructions for each of the five steps. This same process and scripts have been successfully utilized in production migration projects.

Download from [this link](https://github.com/microsoft/AzureSynapseScriptsAndAccelerators) and save the zipped file “**AzureSynapseScriptsAndAccelerators-main.zip**” to workstation or laptop.

Graphical user interface, text, application, email

Description automatically generated

Figure 1 Download Migration Tools from Github

Unzipped (Extract All) the downloaded into a designated directory with the default folder name: “**APS to SQL DW Migration – Schema and Data Migration with Polybase**”. This folder contains all the scripts utilized by the POC.

A screenshot of a social media post

Description automatically generated

Figure 2 Migration Tools File Structure

You can move this “**APS to SQL DW Migration – Schema and Data Migration with Polybase**” to any location (directory) of your choice. However, we recommend moving all contents into **C:\APS2QLDW**, as illustrated in Figure 3 below. From there, many of the instructions in [Github](https://github.com/microsoft/AzureDWScriptsandUtilities/tree/master/APS%20to%20SQL%20DW%20Migration%20-%20Schema%20and%20Data%20Migration%20with%20PolyBase) can be directly applied. In addition, the configuration files (.csv) will need minimal modifications. We also have job-aid powershell scripts to help automate the configuration file generation process (for example, step 4: Generate\_Step4\_ConfigFile.ps1).

A screenshot of a social media post

Description automatically generated

Figure 3 Recommended Location for Migration Tools

## Setup PowerShell Scripts Permissions

### Option 1

When you start PowerShell on a computer for the first time, the **Restricted** execution policy (the default) is likely to be in effect. The **Restricted** policy does not permit any scripts to run.

To find the effective execution policy on your computer (PowerShell): Get-ExecutionPolicy -list

To run unsigned scripts that you write on your local computer and signed scripts from other users, start PowerShell with the Run as Administrator option and then use the following command to change the execution policy on the computer to **RemoteSigned**:

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine

To run an unsigned script, use the Unblock-File cmdlet (Unblock-File -Path .\fileName.ps1). For more information, please refer to this PowerShell documentation page (link [here](https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_signing?view=powershell-6)).

If you are using Visual Studio Code to run PowerShell Scripts, you need to run the VSC as administrator in order to change Policy.

### Option 2

To run unsigned scripts that you write on your local computer and signed scripts from other users, start PowerShell with

Set-ExecutionPolicy Unrestricted

You can also pin Powershell IDE to taskbar, then Shift + right-click to Run with Elevated Rights and then run below script:

Set-ExecutionPolicy Unrestricted

### Install PowerShell Modules

**Install sqlserver and ImportExcel modules for Powershell**

From pinned Powershell Shift Right-Click Run with Elevated Rights

Install-Module sqlserver

Install-Module -Name ImportExcel

# Implementation Steps and Output

This section describes the implementation steps and output of each step.

All the migration scripts with customer configurations to deliver the output will be delivered to customer.

For more detailed instructions for each of the six steps, please visit [GitHub](https://github.com/microsoft/AzureDWScriptsandUtilities/tree/master/APS%20to%20SQL%20DW%20Migration%20-%20Schema%20and%20Data%20Migration%20with%20PolyBase).

## Step 1: Obtain APS DDLs and DMLs

* + **Step 1A**: Prepare Configuration Files to run migration tools step 1, create APS MPP scripts
  + **Step 1B**: Run Create MPP Scripts.
  + **Output**: T-SQL scripts for APS objects (DDLs and DMLs)

For more detailed instructions for this step, please this GitHub Page: [Step 1 Instructions.](https://github.com/microsoft/AzureDWScriptsandUtilities/tree/master/APS%20to%20SQL%20DW%20Migration%20-%20Schema%20and%20Data%20Migration%20with%20PolyBase/1_CreateMPPScripts)

## Step 2: Format APS DDLs and DMLs

* + **Step 2**: Format (Clean up) APS Scripts.
  + **Input**: Output from Step 1
  + **Output**: Cleansed T-SQL Scripts for APS objects (DDLs and DMLs)

For more detailed instructions for this step, please this GitHub Page: [Step 2 Instructions](https://github.com/microsoft/AzureDWScriptsandUtilities/tree/master/APS%20to%20SQL%20DW%20Migration%20-%20Schema%20and%20Data%20Migration%20with%20PolyBase/2_CleanScripts).

## Step 3: Translate APS DDLs and DMLs

* + **Step 3**: Translate APS DDLs and DMLS into Azure Synapse Analytics T-SQL Scripts.
  + **Input**: Output from Step 2
  + **Output**: T-SQL Scripts for Azure Synapse Analytics

For more detailed instructions for this step, please this GitHub Page: [Step 3 Instructions](https://github.com/microsoft/AzureDWScriptsandUtilities/tree/master/APS%20to%20SQL%20DW%20Migration%20-%20Schema%20and%20Data%20Migration%20with%20PolyBase/3_ChangeSchemas).

## Step 4: Create Export and Import T-SQL Scripts

* + **Step 4**: Create T-SQL Scripts for Exporting APS tables and Importing the same into Azure Synapse Analytics
  + **Input**: Output from Step 3
  + **Output**: Export and Import T-SQL Scripts

For more detailed instructions for this step, please this GitHub Page: [Step 4 Instructions](https://github.com/microsoft/AzureDWScriptsandUtilities/tree/master/APS%20to%20SQL%20DW%20Migration%20-%20Schema%20and%20Data%20Migration%20with%20PolyBase/4_CreateAPSExportScriptSQLDWImportScript).

## Step 5: Create T-SQL Scripts for Azure Synapse External Tables

* + **Step 5**: Create T-SQL Scripts for Azure Synapse Analytics External Tables (which point to location of data stored in Azure Storage)
  + **Input**: Output from Step 3
  + **Output**: T-SQL Scripts for Azure Synapse Analytics External Tables (which point to location of data stored in Azure Storage)

For more detailed instructions for this step, please this GitHub Page: [Step 5 Instructions](https://github.com/microsoft/AzureDWScriptsandUtilities/tree/master/APS%20to%20SQL%20DW%20Migration%20-%20Schema%20and%20Data%20Migration%20with%20PolyBase/5_CreateExternalTablesSQLDW).

## Step 6: Deployments

This step performs tasks using outputs from previous steps 1-5. The overall output of these deployment steps: DDLs and DMLs created in ASA. External Tables are created in ASA. Tables are exported from APS to Blob Storage, and then imported from Blob Storage to ASA.

The tasks can be broken down into below groups:

* + **Step 6A**: Run T-SQL Scripts to create Tables, Views, and Stored Procedures in Azure Synapse Analytics
  + **Input**: Output from Step 3
  + **Results of 6A**: Tables, Views, and Stored Procedures created in Azure Synapse Analytics
  + **Step 6B:** Run T-SQL Scripts to create external tables in Azure Synapse Analytics
  + **Input**: Output from Step 5
  + **Results of 6B**: External Tables created in Azure Synapse Analytics
  + **Step 6C**: Run APS Export Scripts
  + **Input**: Output from Step 4
  + **Results of 6C**: APS data exported to Azure Storage
  + **Step 6D**: Run Azure Synapse Analytics Import Scripts
  + **Input**: Output from Step 4
  + **Results of 6D**: Data is imported into Azure Synapse Analytics

For more detailed instructions for this step, please this GitHub Page: [Step 6 Instructions](https://github.com/microsoft/AzureDWScriptsandUtilities/tree/master/APS%20to%20SQL%20DW%20Migration%20-%20Schema%20and%20Data%20Migration%20with%20PolyBase/6_DeployScriptsToSqldw).

## Spot check Table and stored procedures and resolve issues

* + **Spot Check 2 Tables**: Check # of records, check contents, check distributions, partitions.
  + **Spot Check 2 Views**: Compile and execute T-SQL scripts from the views
  + **Spot Check 2 Stored Procedures**: Compile and run the stored procedures with default values

## POC Results Verification Tests

* + **POC Results Verification Tests**: Customer team will test Azure Synapse Analytics with T-SQL Scripts to verify the POC results. Customer team will document issues/errors and feedback to CSE team.

# Migration Exceptions Handling

This section provides tips on exceptions handlings. Not all issues may arise in your migration project. If they do, these tips will provide additional help.

## Additional Code Conversion

If your code is imbedded in configuration tables, the m

Table: EtlJobsAction

|  |  |  |
| --- | --- | --- |
| RecordID | Condition | CodeToExecute |
| 1 | Condition\_A | … [ApsDbName].sys.tables … |
| 2 | Condition\_B | … [APsDbName].sys.views … |
| 8 | Condition\_X | … SET @Msg = 'Droped the table ' + @schema + '.' + @tablename + '\_new if existed' … |

The code containing “… [ApsDbName].sys.tables …” needs to be replaced with “ … sys.tables …”. The code containing “… [ApsDbName].sys.views … “ needs to be replaced with “… sys.views …”. For a complete list of examples like these, please review to

## Step 6: Deployments

# Appendix T-SQL Scripts (Samples)

This section provides sample T-SQL Scripts that are used to set up APS and Azure Synapse Analytics environments. For security reasons, things like server name, database name, azure storage keys are made up values (samples). Users of these scripts will need to replace them with actual values to run these T-SQL Scripts.

All the T-SQL Scripts can be found in this attached zipped file: T-SQL-Scripts-Samples.zip.

