



APS/PDW to Azure Synapse Analytics migration

AMA Group

AMAArchitects@service.microsoft.com

Agenda

- Migration Approach
- APS environment assessment
- APS vs Azure Synapse
- Migration Process and Options
- Migration Tools



Migration Approach

APS Migration – Goals and Benefits

- Goals:
 - Smooth and effective deployment to achieve customer success
 - Set up implementation work for success
 - Drive Partner business opportunities
 - Drive deployment of Azure Synapse
 - Drive quality references
- Achieved through:
 - Quality knowledge transfer to customers
 - Customer specific discovery and design
 - Based on the customer's business requirements
 - Deliver an implementable plan
 - Provide migration tools

APS Migration

Assessment & Discovery:

- a) Complete a migration assessment questionnaire.
- b) Perform an Assessment of the APS appliance (tools available from MS).
- c) Determine Database to Schema migration approach.
- d) Determine the scope of the migration.

Implementation:

- a) Configure Export/Import Process (APS & Synapse)
- b) Script out the APS objects (Schemas/Tables/Views/SPs).
- c) Get Data Validation – Row Counts on Tables in APS
- d) Create Synapse Schemas/Tables/Views
- e) Create/Migrate Synapse SPs
- f) Create External Tables (APS and Synapse)
- g) Setup Scheduled Jobs (ETL) (Carried out by Customer)
- h) Export Data to Azure
- i) Import Data into Synapse
- j) Rebuild Indexes/Stats
- k) Migrate Security (Carried out by Customer with Microsoft's Assistance)
- l) Validate Data between APS and Synapse
- m) Performance tune Synapse

APS environment assessment

Discovery Workshop - Objectives

- Provide customer with overview of how to migrate existing DW solutions to Azure
- Review the high-level vision of the customer, business, and project scope
- Gather detailed metadata on the existing APS system to accurately estimate the migration
- Understand data use cases
- Understand end-to-end solution architecture
- Understand availability requirements of data
- Understand load / query performance expectations
- Identify 3rd party solution integration tools in use
- Understand code artifacts requiring migration re-work in the existing solution
- Understand data flow timing and the available windows for migration
- Understand the existing infrastructure of the complete solution

APS Migration Steps – Discovery workshop

The discovery workshop duration depends on the complexity of the customers implementation.

1. Complete a migration assessment questionnaire. – **Prior to workshop**
2. Perform an Assessment of the APS System (tools available from Microsoft) - **Prior to workshop**
3. Determine Database to Schema migration approach.
4. Determine the scope of the migration.

APS Migration Steps – Questionnaire

Document is prepared by both you and the customer to help determine the correct Azure infrastructure needed to support their existing and future plans.

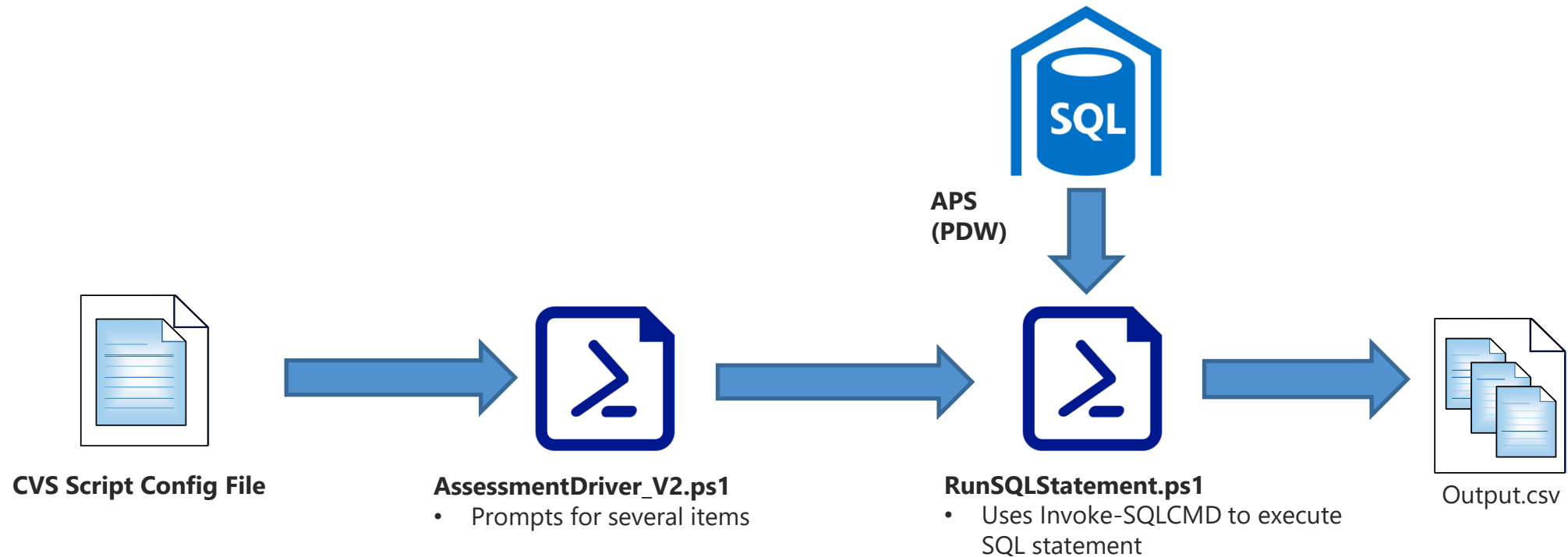
APS Questionnaire		
Hardware Features	Questions	Answers
	How many scale units of PDW?	
	How many scale units of HDI?	
	Is there a Loading Server? If so, is it connected via Infiniband?	
	How much storage is available(Free Space) on the loading Server?	
	Are other servers connected to the Infiniband? If so, What are they used for?	
APS Environments		
	Describe the environments you are using today?	Dev/QA/Prod/DR
	Do you have a Rack for each Env? If not, What environments are sharing hardware?	
	Are all the Prod DB's located on the Prod APS?	
	Provide a Environment Architecture diagram.	
	Provide a data flow diagram.	
	Do you use an APS monitoring utility(DW insights Dashboard)\SCOM\Sentry? If so, what utility are you using?	
Azure Environment		
	Do you currently use Azure? For Production?	
	If using Azure, What services? IAAS/PAAS Services?	
	Do you have an Express Route in place? Bandwidth?	
	Does your APS contain PII Data?	
	If PII\PCI\HIPPA.... Data, Has your security group certified Azure for PII data? If so, what services in Azure? What regulatory requirements do you have for your data?	
	Do you currently perform charge backs\cross charging for APS usage?	
	Are you replicating AD to Azure?	

APS Migration Steps – Assessment

Assessment script to gather information on the APS system to better enable an accurate estimate for the migration. This script captures the following info:

1. Version of the APS system – @@version - Version_{Datetime}.csv
2. Count of all objects in all APS databases. – sys.objects - ObjectCount_{Datetime}.csv
3. List of all tables and their attributes (distribution type, # partitions, storage type and Distribution column) – various system tables – TableMetadata_{DateTime}.csv
4. Listing of the ShowSpaceUsed for all tables. – DBCC pdw_showspaceused – ShowSpaceUsed_{Datetime}.csv
5. Report the number of nodes and total number of distributions on the APS.
6. List of Tables with table name, schema name, and database name in a CSV file that can be used to script out the "Create Table" Statements.
7. List of Views with view name, schema name, and database name in a CSV file that can be used to script out the "Create View" Statements.
8. List of Stored Procedures with stored procedure name, schema name, and database name in a CSV file that can be used to script out the "Create Proc" Statements.

APS Migration Steps – Assessment



1 = Run the line
0 = Skip the line

Active	SourceSystem	RunFor	DB	CommandType	VersionFrom	VersionTo	ExportFileName	ScriptName
1	APS	DB	ALL	SQL	1.0.0.0	99.0.0.0	ShowSpaceUsedTotal	\\APS\ShowSpaceUsedTotal_V1.sql
1	APS	Server	None	SQL	1.0.0.0	99.0.0.0	Version	\\APS\Version_V1.sql
1	APS	Server	None	SQL	1.0.0.0	99.0.0.0	sp_configure	\\APS\sp_configure_V1.sql
1	APS	Server	ALL	SQL	1.0.0.0	99.0.0.0	Distributions	\\APS\Distributions_V1.sql
1	APS	DB	ALL	SQL	1.0.0.0	99.0.0.0	ObjectCount	\\APS\ObjectCount_V1.sql
1	APS	DB	ALL	SQL	1.0.0.0	99.0.0.0	TableMetadata	\\APS\TableMetadata_V1.sql
1	APS	DB	ALL	SQL	1.0.0.0	99.0.0.0	TablesToScript	\\APS\TablesToScript_V2.sql

Migration Tools Steps – Assessment Config

Modify configuration CSV file for AssessmentDriver_V2.ps1 script.
This is a CSV file containing the following columns:

Parameter	Purpose	Sample Value
Active	1 – Run line, 0 – Skip line	0 or 1
SourceSystem	APS, Netezza, Snowflake, Oracle, Teradata	APS
RunFor	Server – the query will be executed at server level DB – the query will be executed at database level Table – the query will be executed at tables level	ALL
DB	ALL – the query will be executed for all databases None – the query will be executed for server instance only	ALL
CommandType	SQL – the query is SQL-query DBCC – the query is DBCC command	SQL
ExportFilename	Name of the output file to save the sql statement results. Datetime will be added to the end of the filename along with .csv.	ObjectCount
ScriptName	Name of the file to store the script in. This should match the object name if possible	ObjectCount_V1.sql

Migration Tools Steps – Assessment

Run PowerShell script AssessmentDriver_V2.ps1. This script will prompt for the following information:

- **“Enter the name of Source System Type to connect to.”** – This will be **APS**.
- **“Would you like to filter the Database to Inventory.”** – Leave as ‘%’ to collect inventory for all APS database.
- **“Enter the name/ip of the Server to connect to.”** – This is the name or IP-address of your target APS appliance (control node).
- **“How do you want to connect to the DB.”** – This is authentication type to be used when connecting to APS. Not that APS support **sqlauth** and **WinInt** only.
- If you chose sqlauth (SQL authentication)
 - **“Please enter the username”** – Username with permission run the scripts (sysadmin role is recommended).
 - **“Password”** – Enter the Password for the user – reads password as a secure string

APS vs Azure Synapse

Architecture for APS - Distributions

CREATE TABLE myTable (column Defs) WITH (DISTRIBUTION = HASH (id));

MPP Node 1

Create Table <table GUID>_a
Create Table <table GUID>_b
...
Create Table <table GUID>_h

} 8 tables per node

MPP Node 2

Create Table <table GUID>_a
Create Table <table GUID>_b
...
Create Table <table GUID>_h

MPP Node ...

MPP Node 8

Create Table <table GUID>_a
Create Table <table GUID>_b
...
Create Table <table GUID>_h

HP

7 - Racks

2 - 56 Compute nodes

8-node (1 data rack) appliance

Dell & Quanta

6 - Racks

2 - 54 Compute nodes

9-node (1 data rack) appliance

Final result – 8 Compute Nodes

64 individual tables across an

8-node (1 data rack) appliance

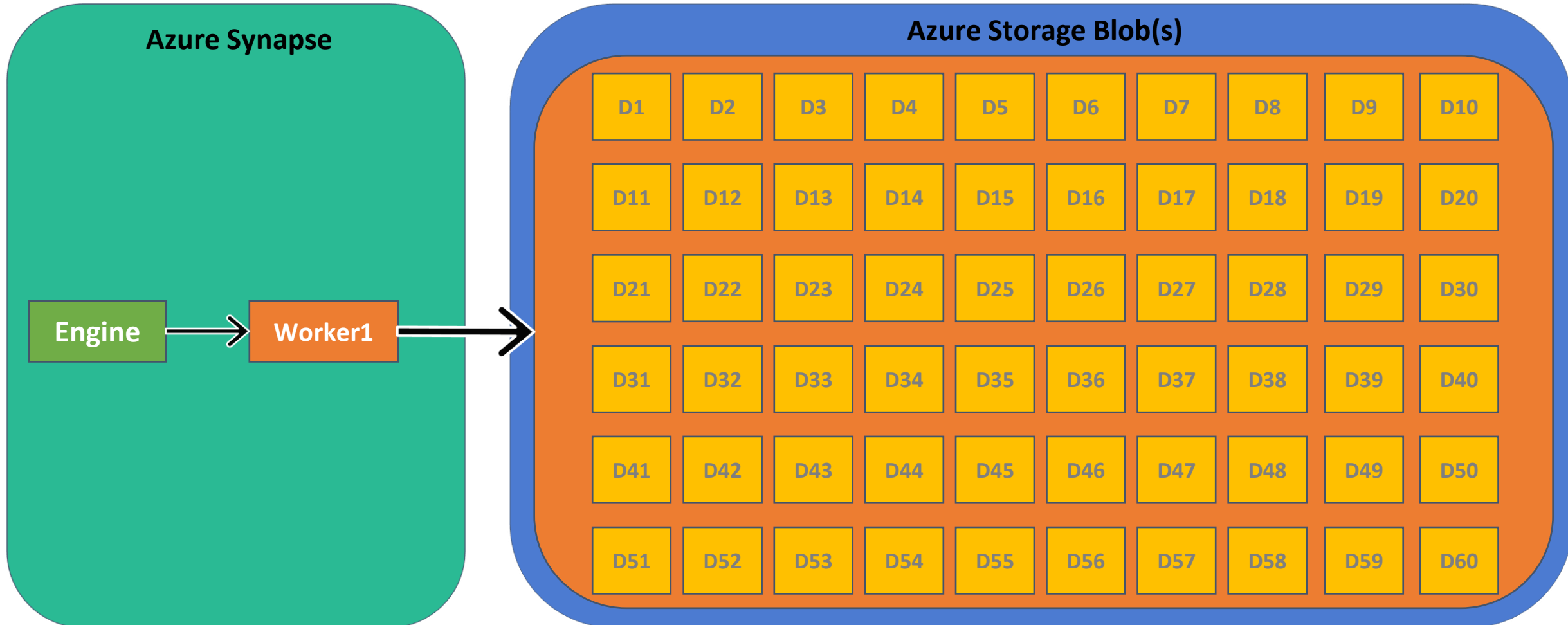
Architecture for Azure Synapse - Distributions

CREATE TABLE myTable (column Defs) WITH (DISTRIBUTION = HASH (id));

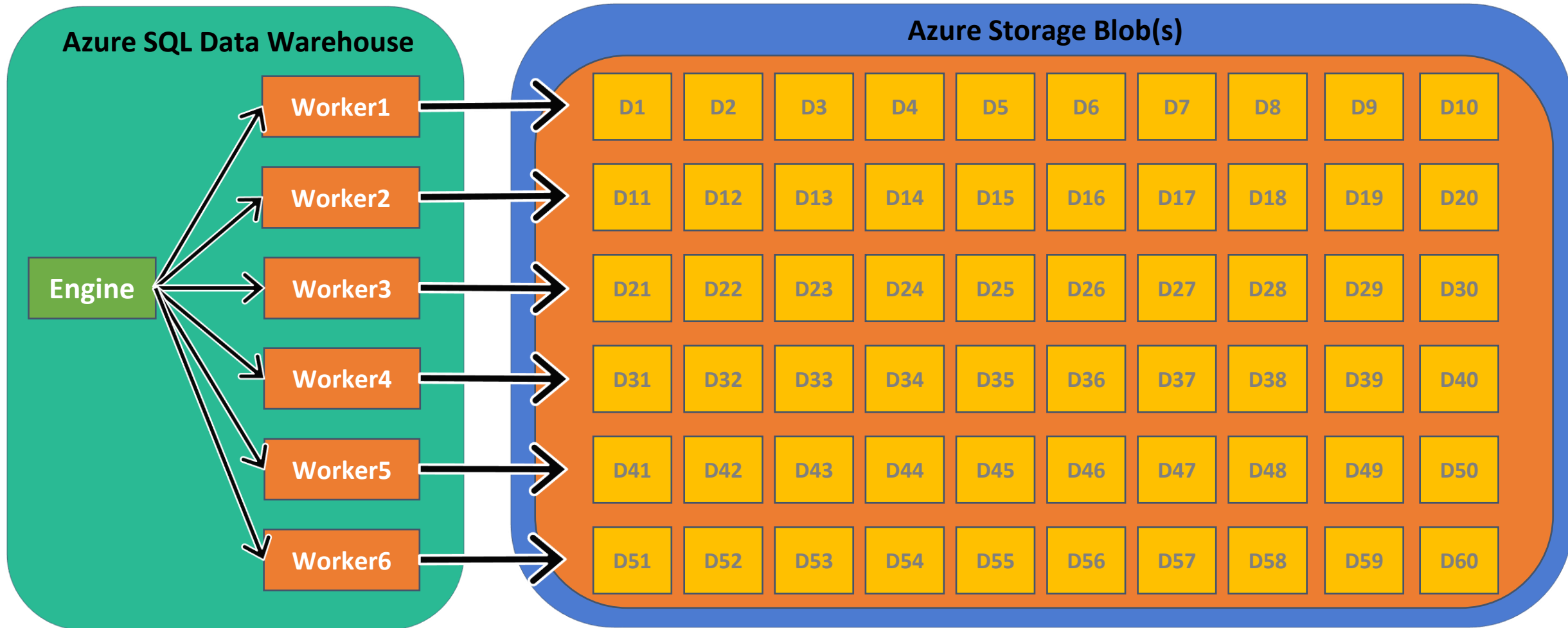
Azure Storage Blob(s)

D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
D11	D12	D13	D14	D15	D16	D17	D18	D19	D20
D21	D22	D23	D24	D25	D26	D27	D28	D29	D30
D31	D32	D33	D34	D35	D36	D37	D38	D39	D40
D41	D42	D43	D44	D45	D46	D47	D48	D49	D50
D51	D52	D53	D54	D55	D56	D57	D58	D59	D60

Architecture for Azure Synapse – DWU100



Architecture for Azure Synapse – DWU3000



APS Migration – APS & Synapse Major Differences

APS:

- Compute nodes range from 2 – 56
- Distributions vary from 16 – 448
- SQL Capabilities depend on version installed
- GetDate() – Returns date based on APS time zone
- Multiple Databases
- Default Index – HEAP
- DWLoader

Azure DW:

- Compute nodes range from 1 – 60
- Distributions are Static at 60
- SQL Capabilities - What's currently available on Azure SQL DW
- Getdate() – Returns only UTC Time. Can't be changed
- Single Database
- Default Index – Clustered Column Store
- **No DWLoader - SQL DW Upload Task in SSIS (SSIS Azure Feature pack)**

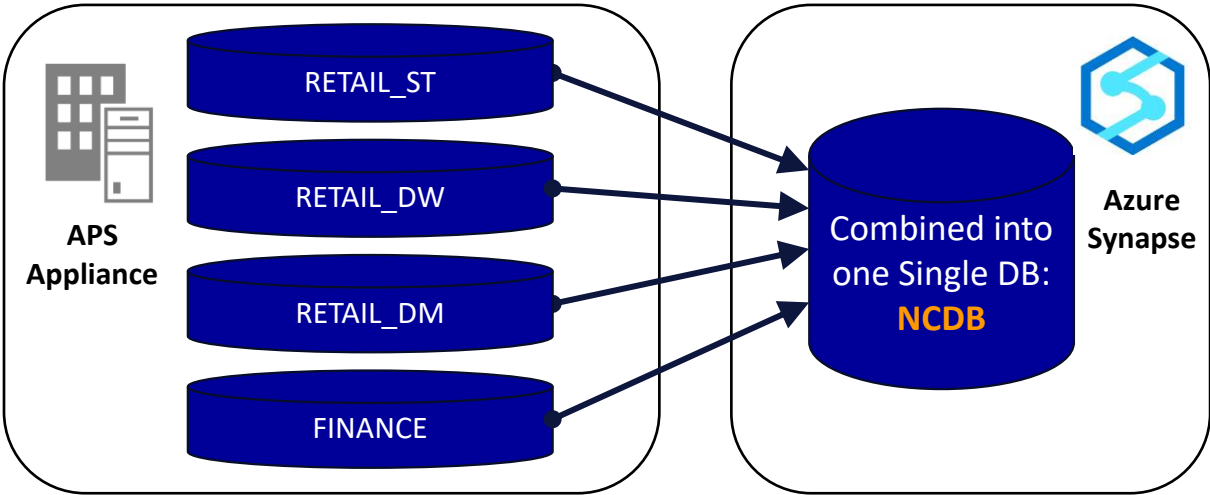
DDL Migration - DB based vs. Schema Based

APS is an instance level service that supports multiple databases. **Azure Synapse is a database-level service that has one database per instance.**

In order to migrate APS code and data into Synapse, DB name and schema name must be changed.

An example is shown on the right to illustrate the changes.

After the new DB name and schema names are determined, all meta data (DDLs & DMLs: Tables, Views) and code (Stored Procedures) must be modified accordingly. Then the new DDLs, DMLs, and SPs can be installed in Synapse.



APS Database	APS Schema Name	Synapse DB Name	Synapse Schema Name
RETAIL_ST	DBO	NCDB	RETAIL_ST_DBO
RETAIL_DW	DBO	NCDB	RETAIL_DW_DBO
RETAIL_DM	DBO	NCDB	RETAIL_DM_DBO
FINANCE	FN	NCDB	FINANCE_FN

Migration Process & Options

Migration Process - APS

The following items need to be migrated from APS to Synapse.

1. DDL (Table)
2. Code (Stored Procedures, Views and ETL process)
3. Security
4. Data

The DDL and DML can be migrated using internal tools available from Microsoft (APS Migration Tools).

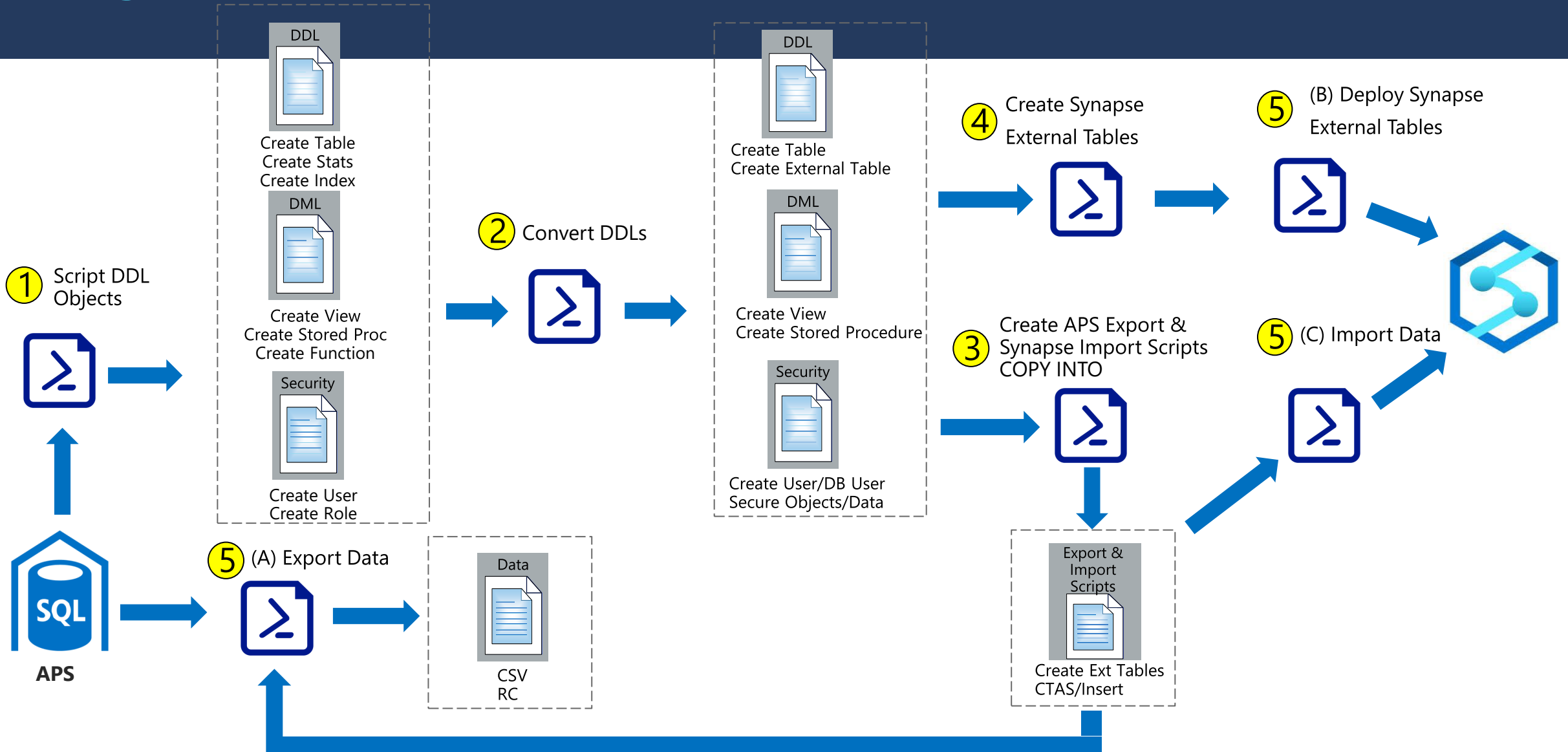
Security will need to be migrated by hand.

To migrate data to Azure, several approaches can be taken. Each approach has a set of prerequisites as well as pros and cons. Each method should be analyzed considering customers environment, data load frequency and security requirements.

Data Migration approaches:

1. BCP data out of APS – Polybase data into Synapse from WASB or Azure Data Lake
2. Polybase data out of APS to Azure Storage Blob - Polybase data into Synapse from ASB or Azure Data Lake.
3. SSIS – Source(APS) Destination(Synapse)
4. ADF – Extract data from APS to Synapse
5. Import Export Service

Migration Tasks



Data Migration Approach - BCP

BCP Data out of APS – Polybase data into Synapse

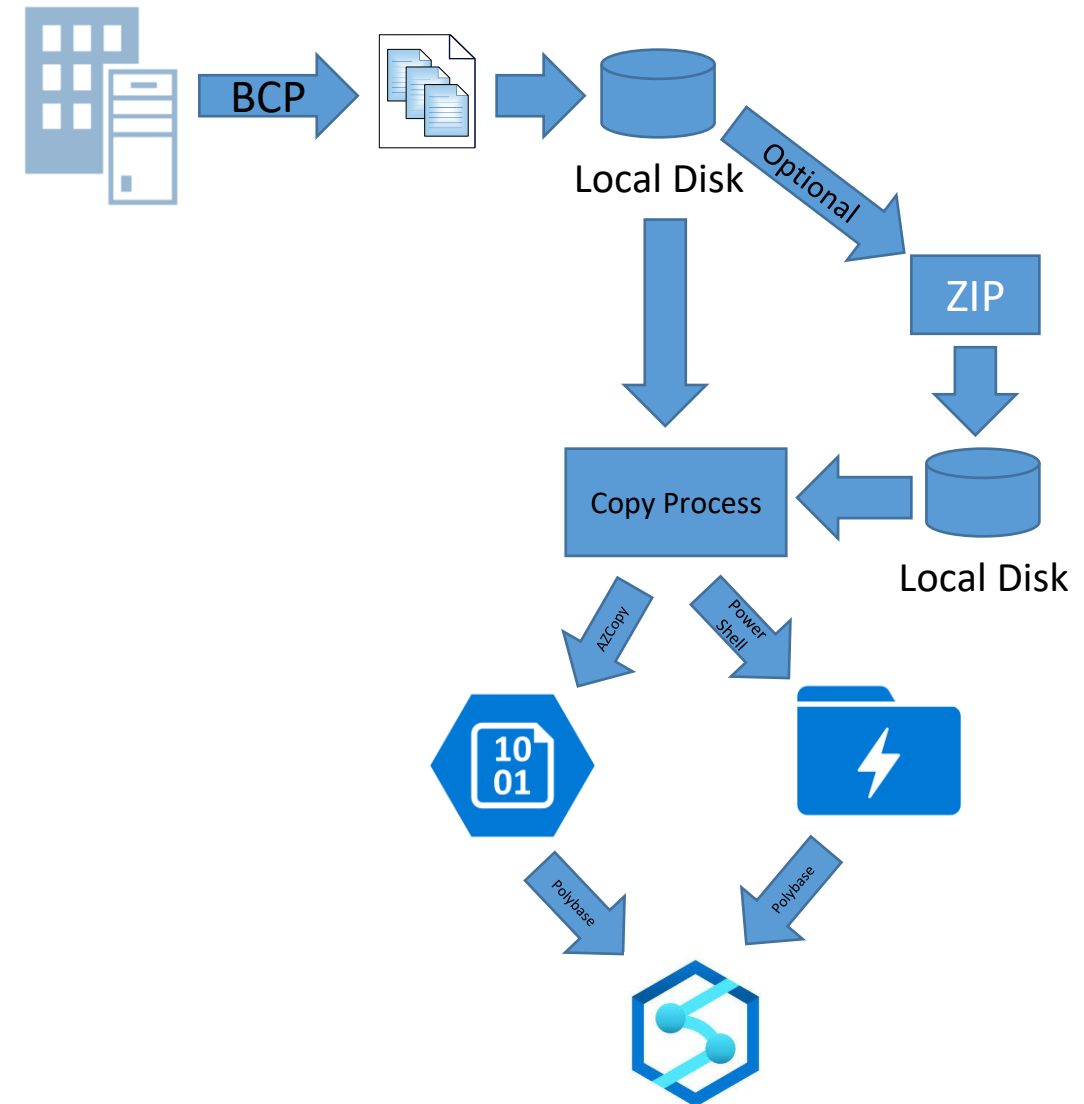
While APS did not receive BCP in support until AU4, this method will still work on earlier versions as the process is exporting data and not importing. This process utilizes the SQL Server tool BCP to export data out of the APS system and store the data in Flat Files. These files are then copied to either WASB or Azure DataLake where they can be consumed using Polybase to load Synapse.

Pros:

- Can break up large table by partition.
- Can Zip files before moving them to Azure in another step.
- Familiar tool for SQL Server DBA's.
- Data reloads for Dev/QA/Prod does not need to extract unchanged data from APS

Cons:

- All data is passed through the control node and therefore is single threaded. To utilize multiple export paths, multiple BCP commands will be necessary.
- Must have local disk space to store files until they are moved to Azure.
- Potential file delimiter issues



Data Migration Approach – Polybase Out Polybase In

Polybase Data out of APS – Polybase data into Synapse

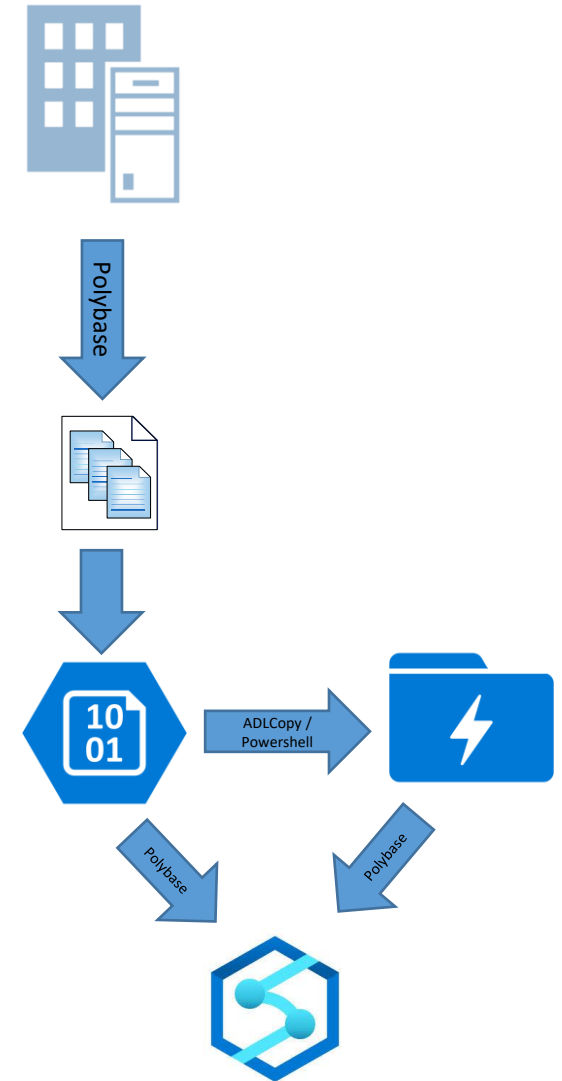
Polybase is one of the fastest methods for importing and exporting data from APS to Synapse. However, the true speed is dependent on the connection speed between APS and Azure. The overall speed needs to be tested. Before using this method, clearance should be obtained from the organizations IT/Security group to ensure the export will not consume the bandwidth of the Azure connection. Polybase does have the ability to compress files as they are created. APS can only create the Polybase file on WASB. If they are to be stored/consumed from ADLS, an additional step will be needed to copy them to ADLS from WASB.

Pros:

- Can break up large table by partition. Will require additional External Tables.
- Can Zip files before moving them to Azure.
- Each Distribution runs in parallel making this the fastest method to export/import file into Synapse. Export speed is dependent on the network connection to Azure.
- Data reloads for Dev/QA/Prod do not need to extract unchanged data from APS again

Cons:

- APS can only utilize WASB currently.
- Potential file delimiter issues



Data Migration Approach – SSIS

SSIS – ELT of data from APS to Synapse

SSIS can be used to automate the data extract from APS and import into Synapse.

There are two (2) Options with this approach:

- 1) Extraction of the data to flat files in one task. Utilize Polybase to import the Files to Synapse
- 2) Extraction of the data and direct insert into the Synapse Tables.

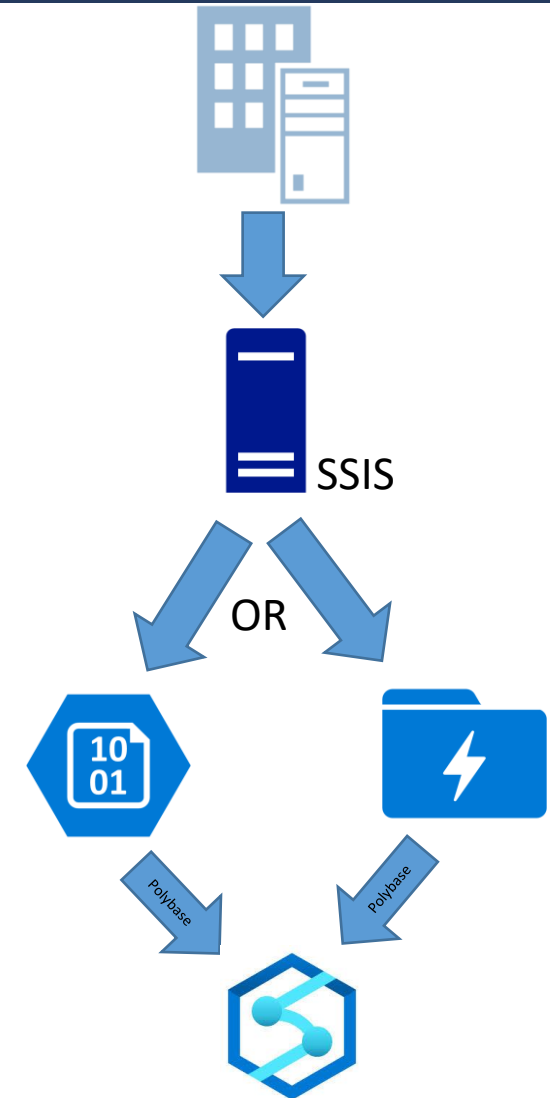
Flat File to Azure – Polybase into Synapse Approach:

Pros:

- Can Zip files before moving them to Azure.
- Large Tables can be extracted by partition eliminating the need to load all the data in a single migration operation.
- Can move the File to WASB or ADLS
- Logging built into SSIS
- Can use Scripting to create SSIS Packages (BIML, Ezapi, SSIS API)
- Data reloads for Dev/QA/Prod do not need to extract unchanged data from APS

Cons:

- Additional overhead of using SSIS (Licenses and Server)
- Possible delimiter issues



Data Migration Approach – SSIS

SSIS – ELT of data from APS to Synapse

Direct Insert to Synapse:

Pros:

- Large Tables can be extracted by partition eliminating the need to load all the data in a single migration operation.
- Logging built into SSIS
- Can use Scripting to create SSIS Packages (BIML, Ezapi, SSIS API)
- No need for additional space for flat Files
- No issues with delimiters

Cons:

- Data reloads during testing must extract data from source.
- Additional overhead of using SSIS (Licenses and Server)
- Data is not compressed before sending to Synapse



Data Migration Approach – ADF

ADF – Data extract from APS to Synapse

ADF can be used to automate the data extract from APS and import into Synapse.

There are two (2) Options with this approach:

- 1) Extraction of the data to flat files in one task. Utilize a second task to import the Files to Synapse
- 2) Extraction of the data and direct insert into the Synapse Tables.

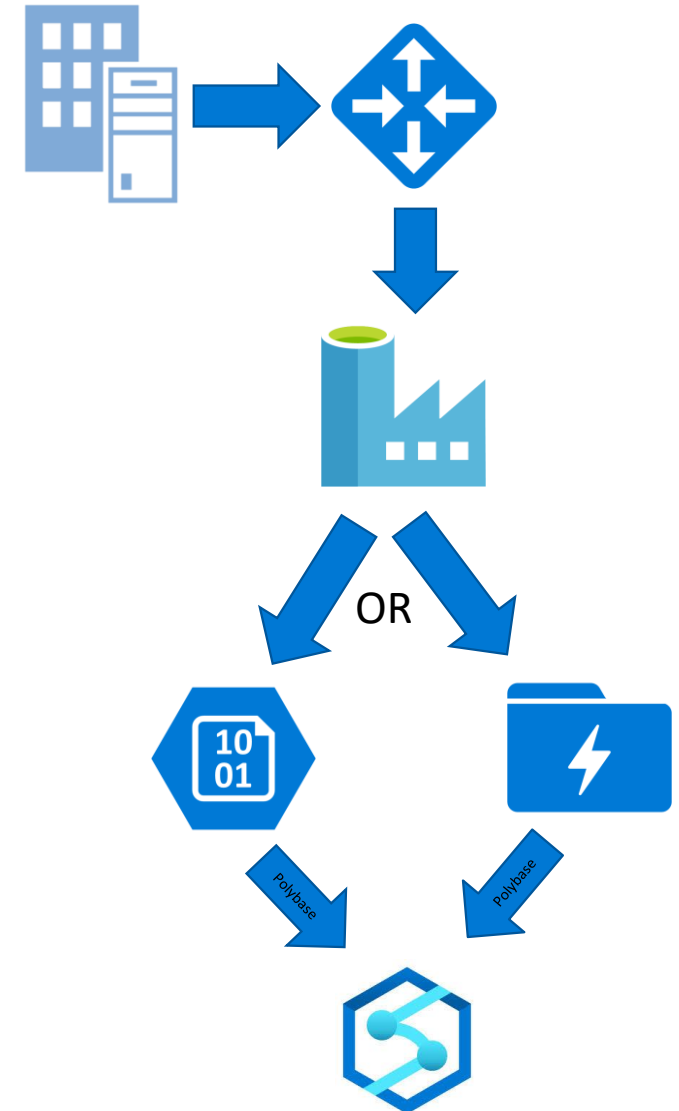
Flat File Extract to Blob/ADLS Approach:

Pros:

- Large Tables can be extracted by partition eliminating the need to load all the data in a single migration operation.
- Can move the File to WASB or ADLS
- Logging built into ADF
- Can use Scripting to create ADF process
- Data reloads for Dev/QA/Prod does not need to extract unchanged data from APS

Cons:

- Possible delimiter issues
- Speed is limited to the speed of the control node. Data is copied to Blob or ADLS first. Does not use Polybase on the APS to extract data.
- Additional setup and configuration of Azure Data Management Gateway
- Files are copied to WASB or ADLS before being sent to Synapse



Data Migration Approach – ADF

ADF – Data extract from APS to Synapse

Direct Insert to Synapse:

Pros:

- Large Tables can be extracted by partition eliminating the need to load all the data in a single migration operation.
- Logging built into ADF
- Can use Scripting to create ADF Tasks

Cons:

- Data reloads during testing must extract data from source.
- Speed is dependent on the how fast the control node can provide data. This does not use polybase to extract data on APS.
- Additional setup and configuration of Azure Data Management Gateway
- Potential Delimiter issues
- Files are copied to WASB or ADLS before being sent to Synapse



Data Migration Approach – Azure Import/Export Service

Azure Export/Import Service

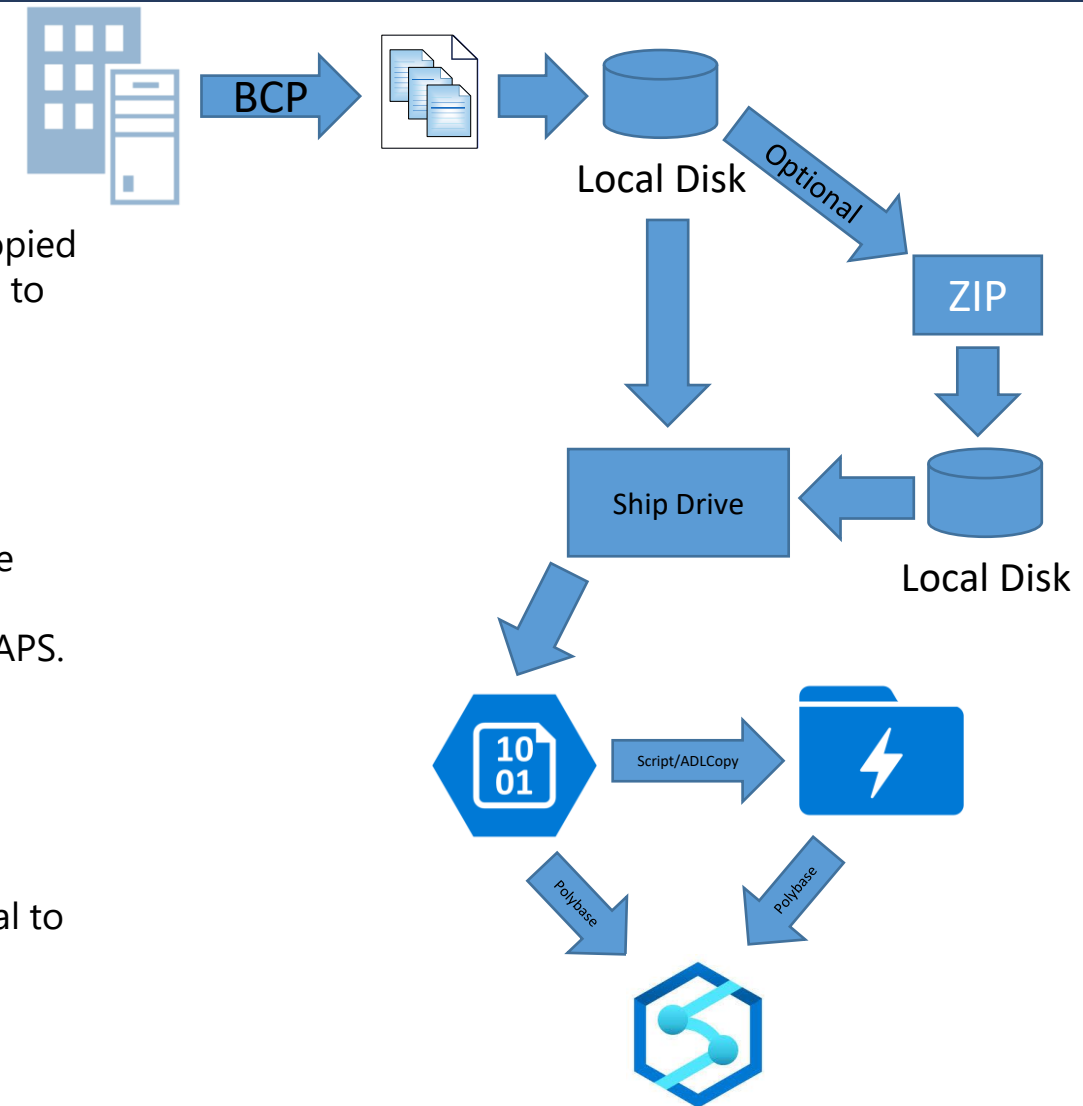
The Azure export/import service can be used to transfer large datasets from on premises to Azure through the use of a SATA Hard Drive. The files need to be copied to an external hard drive that has been registered/configured in the Azure Portal to be used with this service. The drive is then shipped to Azure where the data is copied to the specified WASB account during configuration.

Pros:

- Can Zip files before moving them to azure.
- Large Tables can be extracted by partition eliminating the need to load all the data in a single migration operation.
- Data reloads for Dev/QA/Prod do not need to extract unchanged data from APS.
- Bitlocker encrypted drive.

Cons:

- Additional time to get data to Azure. Shipping and loading process.
- Possible delimiter issues.
- Changes/Deltas between the initial extract and the final load has the potential to be larger than the previous methods.
- Can only load to WASB.
- Additional configuration\install\removal\shipping of SATA Drive.



Data Migration Approach – Azure Data Box Service

Azure Data Box (Disk) Service

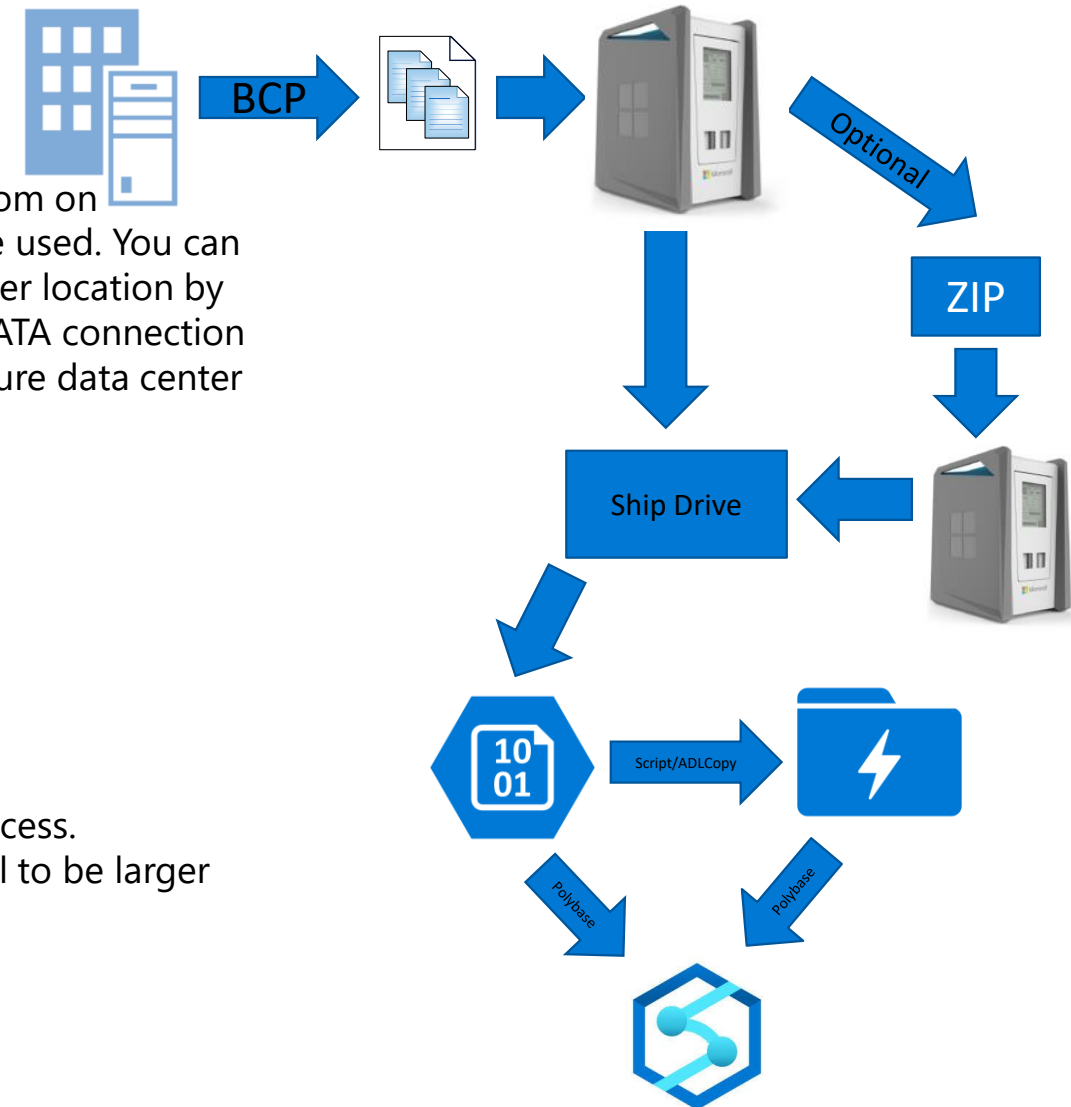
The Data Box service can be used to transfer very large datasets (Up to 100TB) from on premises to Azure. If the data size is less than 40TB, Data Box Disk service may be used. You can order the Data Box or Disk through Azure portal which will be shipped to customer location by Microsoft. Customer can then connect the Data Box into their network (or USB/SATA connection in case of Data Disk) and copy the files to it. The Box or Disks then shipped to Azure data center where the data is imported into customer specified WASB account.

Pros:

- Can Zip files before moving them to azure.
- Very large datasets (40 to 100TB) can be moved easily.
- Microsoft provides the necessary storage hardware.
- AES encryption of the data for strong security.

Cons:

- Additional time needed to get data to Azure due to shipping and loading process.
- Changes/Deltas between the initial extract and the final load has the potential to be larger than the previous methods.
- Relatively less repeatability.
- Additional configuration\install\removal\shipping of storage.




SSIS Migration Approach – Lift and Shift with ADF V2

If customer is using SSIS packages for ongoing ingestion of data, same packages can be migrated to Azure for ingesting data directly into Synapse.


- Provision SSIS on Azure
 - Create SSIS Catalog (SSISDB)
 - Create Azure-SSIS Integration Runtime
- Modify Packages
 - Use latest version of SSDT (15.3 or later)
 - Modify connection strings with SQL DW as target
- Deploy Packages to SSIS on Azure
 - Connect to SSISDB in Azure
 - Use project deployment model
- Run/Schedule Packages
 - Use Execute SSIS Package activity in ADF V2
 - Schedule using
 - ADF Pipeline
 - SSMS
 - SQL MI Agent
 - SQL Server Agent on-premises

Integration Runtime Setup

Integration Runtime is the native compute used by ADF to execute or dispatch activities. Choose what integration runtime to create based on required capabilities.



Perform data movement and dispatch activities to external computes.



Lift-and-shift existing SSIS packages to execute in Azure.

CancelNext →

APS Polybase Setup (pre – APS 2016)

The following preparation steps need to be performed on the APS system. Please work with the Customer's APS support engineer to configure the APS with these steps. This will require a restart of the APS region.

1. Modify core-site.xml file on PDW control node to add Azure BLOB storage credentials
This file is available under **C:\Program Files\Microsoft SQL Server Parallel Data Warehouse\100\Hadoop\conf**
2. Modify 'Hadoop Connectivity' PDW configuration option to enable BLOB connectivity
Ex: `EXEC sp_configure 'hadoop connectivity', 4;`
`RECONFIGURE;`
3. Restart PDW region

APS Polybase Setup (pre – APS 2016) – contd.

After BLOB storage access is configured, the following steps need to be completed for Polybase access to the Blob storage:

1. Configure DNS forwarders with external DNS names on AD01 and AD02 (if not already done)
2. Create External Data Source (Linked to the Azure Blob Storage Account and a particular container)

```
CREATE EXTERNAL DATA SOURCE myBlobDS with (  
    TYPE = HADOOP,  
    LOCATION = 'wasbs://myblobcontainer@myblobstorageaccount.blob.core.windows.net',
```
3. Create External File Format

```
CREATE EXTERNAL FILE FORMAT RCFileFormat WITH (  
    FORMAT_TYPE = RCFILE,  
    SERDE_METHOD = 'org.apache.hadoop.hive.serde2.columnar.LazyBinaryColumnarSerDe' )
```

APS Polybase Setup (APS 2016+)

The following preparation steps need to be performed on the APS (2016 or higher) system for Polybase access to the Blob storage.

1. Modify 'Hadoop Connectivity' PDW configuration option to enable BLOB connectivity
2. Create a Master Key on the database (if not already done)
3. Create Database Scoped Credential (linked to Azure Blob Storage Account)
4. Create External Data Source (Linked to the Azure Blob Storage Account and a particular container)
5. Create External File Format

APS Polybase Setup (APS 2016+) – contd.

```
EXEC sp_configure 'hadoop connectivity', 4;  
RECONFIGURE;
```

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'P@ssword!23';
```

```
CREATE DATABASE SCOPED CREDENTIAL myBlobCred
```

```
    WITH IDENTITY = 'myblobstorageaccount', -- Blob Storage Account Name
```

```
    SECRET = 'ThisIsMaskedKeyLongNameLongStringGenerated GetThisKeyFromYourAzureStorageAccount==' --  
            Blob Storage Access Key
```

```
CREATE EXTERNAL DATA SOURCE myBlobDS with (
```

```
    TYPE = HADOOP,
```

```
    LOCATION = 'wasbs://myblobcontainer@myblobstorageaccount.blob.core.windows.net',
```

```
    CREDENTIAL = myBlobCred -- This name must match the database scoped credential name
```

```
CREATE EXTERNAL FILE FORMAT RCFileFormat WITH (
```

```
    FORMAT_TYPE = RCFILE,
```

```
    SERDE_METHOD = 'org.apache.hadoop.hive.serde2.columnar.LazyBinaryColumnarSerDe' )
```

Data Migration – Export/Import Issues

Data export from APS and/or import into Synapse may fail due to certain Polybase limitations, as explained below:

1. Using Delimited Text File Format – Line Feeds (LF) and/or Carriage Returns (CR) embedded inside a string field.
Polybase treats these characters as first-class citizens and breaks the strings during the export.
 - Workaround 1: export using RCFile/ORC/Parquet File Format
 - Workaround 2: Replace embedded CR/LFs with any other supported characters during the export
2. Using RCFile/ORC/Parquet File Format, blank Values in string column will be implicitly converted to NULL during export.
Export will fail if the string column is defined as NOT NULL. Possible data corruption if customer wants to treat NULLs and blanks distinctly
 - Recommendation: Use Delimited Text File Format.
 - Workaround 1: Clean up data in source DB to remove/replace any unnecessary blank values
 - Workaround 2: Temporarily change the empty strings in source DB to supported values, replace them back to empty strings once import is complete
 - Workaround 3: Temporarily change the Synapse column definition to allow Null. After Import, create new table with original APS table definition and CTAS data over to the new table from imported table.
 - Workaround 4 (Not recommended): Set Reject_Value in external table definition to a higher value so that errored rows can be ignored during the export. Even if export does not throw errors, not all records will be imported.

Data Migration – Export/Import Issues (contd.)

3. Exporting datetime data using RCFile, ORC, or ORC format
Datetime values will be converted to UTC during the import.
 - Workaround 1: Convert the datetime values to [varchar](#) during the export. Convert values back to [datetime](#) during the import to Synapse. Ex: ***SELECT CONVERT(NVARCHAR(25), PostTime, 121) AS PostTime FROM [EXTDatabaseLog]***
 - Workaround 2: Use Delimited Text file format for export/import.
4. Exporting with USE_TYPE_DEFAULT set to TRUE in External File Format Definition.
This may cause unintended data value changes, as it will convert all blank values in string columns to NULLs, numeric/float columns to '0' and in DATETIME columns to '1900-01-01'
 - Recommendation: Set the USE_TYPE_DEFAULT to FALSE and take any appropriate action on the blank values before the export

Data Migration – Export/Import Issues (contd.)

5. Exporting data to Premium BLOB storage or ADLS

External table creation with the location pointing to Premium BLOB storage or ADLS will fail as it is currently not supported

- Workaround 1: Use Standard BLOB storage account as the location
- Workaround 2: Use BCP to export the data to local storage first and use manual methods to upload files to BLOB storage/ADLS

6. Exporting data from very large tables

Export may fail intermittently due to network bandwidth or Java related issues

- Workaround 1: Use BCP to export the data to local storage first and use manual methods to upload files to BLOB storage/ADLS
- Workaround 2: Export the data from large tables in batches (Ex: by partition, by filter condition etc.)

Data Migration – Export and Import Data

Export/Import	Scenarios	DELIMITEDTEXT-NoZip	RCFile-NoZip	ORCFile-NoZip	ParquetFile-NoZip
APS-Export	1 - Column Delimiter	OK	OK	OK	OK
APS-Export	2 - String Delimiter	OK	OK	OK	OK
APS-Export	3 - Line Feed	OK	OK	OK	OK
APS-Export	4 - Empty And Null Strings	OK	Empty strings in "Not Null" columns cause errors. Data cleansing necessary.	Same as RC File	Same as RC File
APS-Export	5 -Dates And Time	OK	Cannot Export Dates with Mixed Formats. Work around: Convert Dates to varchar (Export) and then convert back to Dates (Import)	Same as RC File	Same as RC File
Synapse-Import	1 - Column Delimiter	OK	OK	OK	OK
Synapse-Import	2 - String Delimiter	OK if the right string delimiter is used.	OK	OK	OK
Synapse-Import	3 - Line Feed	Problem - MSG107090 - Error: (Tokenization failed), Error: No closing string delimiter. (Issue #1)	OK	OK	OK
Synapse-Import	4 - Empty And Null Strings	OK	Convert empty string to Null.	Convert empty string to Null.	Convert empty string to Null.
Synapse-Import	5 -Dates And Time	OK	Workaround Method: Dates->Varchar->Dates Conversion	Workaround Method: Dates->Varchar->Dates Conversion	Workaround Method: Dates->Varchar->Dates Conversion

* All tests performed with APS 2016 and SQLDW Gen1. Results may differ for older/different APS appliances.

APS to Synapse Conversion – Stored Proc conversion

Description	Before	After	Notes
3-Part and 2-Part Name	[CS_STAGE].[CSS].Table Or CSS.Table	CSS_ST.Table	All 3-part and 2-part naming conventions need to be converted to only contain the schema name. The DB Name has to be removed.
System tables containing DB Name	CS_STAGE.sys.tables	sys.tables	All system tables referenced in a SP will need to have the DB Name removed in Synapse.
System Table Queries	IF EXISTS (SELECT 1 FROM [DBName].SYS.OBJECTS WHERE TYPE = 'U' AND NAME = 'Table_Name')	IF EXISTS (SELECT 1 FROM sys.schemas s JOIN sys.tables t ON s.schema_id = t.schema_id WHERE s.name = 'Schema_Name' AND t.name = 'Table_Name') Or IF EXISTS (SELECT 1 FROM sys.tables t WHERE schema_name(schema_id) = 'Schema_Name' AND t.name = 'Table_Name') Or IF OBJECT_ID('schemaName.TableName, 'U') IS NOT NULL DROP TABLE SchemaName.TableName;	Anytime a system table is referenced and contains a filter to a TableName, the SchemaName should also be included in the query.
Heap Tables (Case by Case)	Create Table With (Distribution = REPLICATE, Location = USER_DB)	Create Table With (Distribution = REPLICATE, HEAP)	If not explicitly specified, Synapse creates Clustered Columnstore Index (CCI) automatically. If CCI is not desirable, perform this change.
Temporary Tables	Create Table #Temp with(Distribution = REPLICATE, Location = USER_DB)	Create Table #temp With(Distribution = ROUND_ROBIN)	Temporary table are not allowed to be created with a replicated distribution. All temporary table need to be changed to permanent tables or the distribution needs to be changed to round_robin Remove Location = User_DB as this is no longer needed.
With Clause with out an Index	With (Distribution = Hash(column_name)) Or With (Distribution = Round_Robin)	With (Distribution = Hash(column_name), HEAP) Or With (Distribution = Round_Robin, HEAP)	Tables created without an index in the with clause on APS need to be converted to explicitly state the index should be a HEAP.
UTC and GetDate()			See Slide on UTC date conversion.

Stored Procedure Change for Temp Tables

Temporary table are not allowed to be created with REPLICATE distribution in Synapse. Temporary table can be changed to a permanent table, or the distribution needs to be changed to ROUND_ROBIN or HASH.

LOCATION = USER_DB – this syntax still works but it makes no sense in Synase, hence, to be removed..

Note: HEAP is added, as the default index on Azure Synapse is CCI (not HEAP).

-- Example SP in AdventureWorks DB after change.

```
CREATE TABLE #TMP_Customers
WITH (DISTRIBUTION = Replicate, Location = User_DB)
WITH (DISTRIBUTION = ROUND_ROBIN, HEAP)
AS
SELECT
.....
```

UTC Date Conversion – GETDATE()

- Synapse returns UTC date and time when using GETDATE() or any system date/time function. Handling this will depend on the client.
- If the Customer would like to store all dates/times in local Time Zone then special handling needs to be performed.
- Create, Update, Update Dates are typically fine, however this will depend on the customer's Time Zone. Any customer in a -1 to -N will be fine. Any customer in a +UTC Time zone will need to review how these are handled. The ETL process will need to be paused for the +hours the customer is ahead of UTC to make sure the next ETL run does not have any overlapping dates and times.
- Stored Calculations like 'Days an order has been open' or 'Days to Fill' will need to be converted to local TZ.
- Run Durations of a SP or a step in a SP can be left as UTC.

Migration Tools Steps – UTC Date Conversion

To Get the local time, the following items need to be completed.

- Create Function to get the local Time

```
CREATE FUNCTION dbo.getdateDST (@UTCDate datetime, @offset varchar(6)) RETURNS datetime
AS
BEGIN
    RETURN convert(datetime, SWITCHOFFSET(CONVERT(DATETIMEOFFSET, @UTCDate), @offset))
END
```

- Create a Table to store the start and end datetimes for DST.

```
CREATE TABLE DSTCalendar
(
    id INT NOT NULL
    , TimeZone varchar(4) NOT NULL
    , Offset VARCHAR(6) NOT NULL
    , OffsetNum TINYINT NOT null
    , FromDate SMALLDATETIME NOT NULL
    , TODate SMALLDATETIME NOT NULL
) WITH (DISTRIBUTION = REPLICATE, HEAP)
```

- Insert record into the DSTCalendar Table

```
INSERT INTO DSTCalendar VALUES (1, 'EST', '-04:00', 4, '3/10/2013 2:00', '11/3/2013 1:59:59')
INSERT INTO DSTCalendar VALUES (2, 'EST', '-05:00', 5, '11/3/2013 2:00', '3/9/2014 1:59:59')
INSERT INTO DSTCalendar VALUES (3, 'EST', '-04:00', 4, '3/9/2014 2:00', '11/2/2014 1:59:59')
INSERT INTO DSTCalendar VALUES (4, 'EST', '-05:00', 5, '11/2/2014 2:00', '3/8/2015 1:59:59')
INSERT INTO DSTCalendar VALUES (5, 'EST', '-04:00', 4, '3/8/2015 2:00', '11/1/2015 1:59:59')
INSERT INTO DSTCalendar VALUES (6, 'EST', '-05:00', 5, '11/1/2015 2:00', '3/13/2016 1:59:59')
INSERT INTO DSTCalendar VALUES (7, 'EST', '-04:00', 4, '3/13/2016 2:00', '11/6/2016 1:59:59')
INSERT INTO DSTCalendar VALUES (8, 'EST', '-05:00', 5, '11/6/2016 2:00', '3/12/2017 1:59:59')
INSERT INTO DSTCalendar VALUES (9, 'EST', '-04:00', 4, '3/12/2017 2:00', '11/5/2017 1:59:59')
INSERT INTO DSTCalendar VALUES (10, 'EST', '-05:00', 5, '11/5/2017 2:00', '3/11/2018 1:59:59')
INSERT INTO DSTCalendar VALUES (11, 'EST', '-04:00', 4, '3/11/2018 2:00', '11/4/2018 1:59:59')
INSERT INTO DSTCalendar VALUES (12, 'EST', '-05:00', 5, '11/4/2018 2:00', '3/10/2019 1:59:59')
INSERT INTO DSTCalendar VALUES (13, 'EST', '-04:00', 4, '3/11/2019 2:00', '11/3/2019 1:59:59')
```

Migration Tools Steps – UTC Date Conversion

- To get the local time, add the following to each SP that needs this functionality.

```
DECLARE @getdate datetime = GETDATE()  
DECLARE @offset varchar(6)  
DECLARE @TimeZone VARCHAR(4) = 'EST' --TimeZone
```

```
SET @offset = (SELECT OffSet FROM dbo.DSTCalendar WHERE @getdate between Fromdate and Todate AND  
Timezone = @Timezone)
```

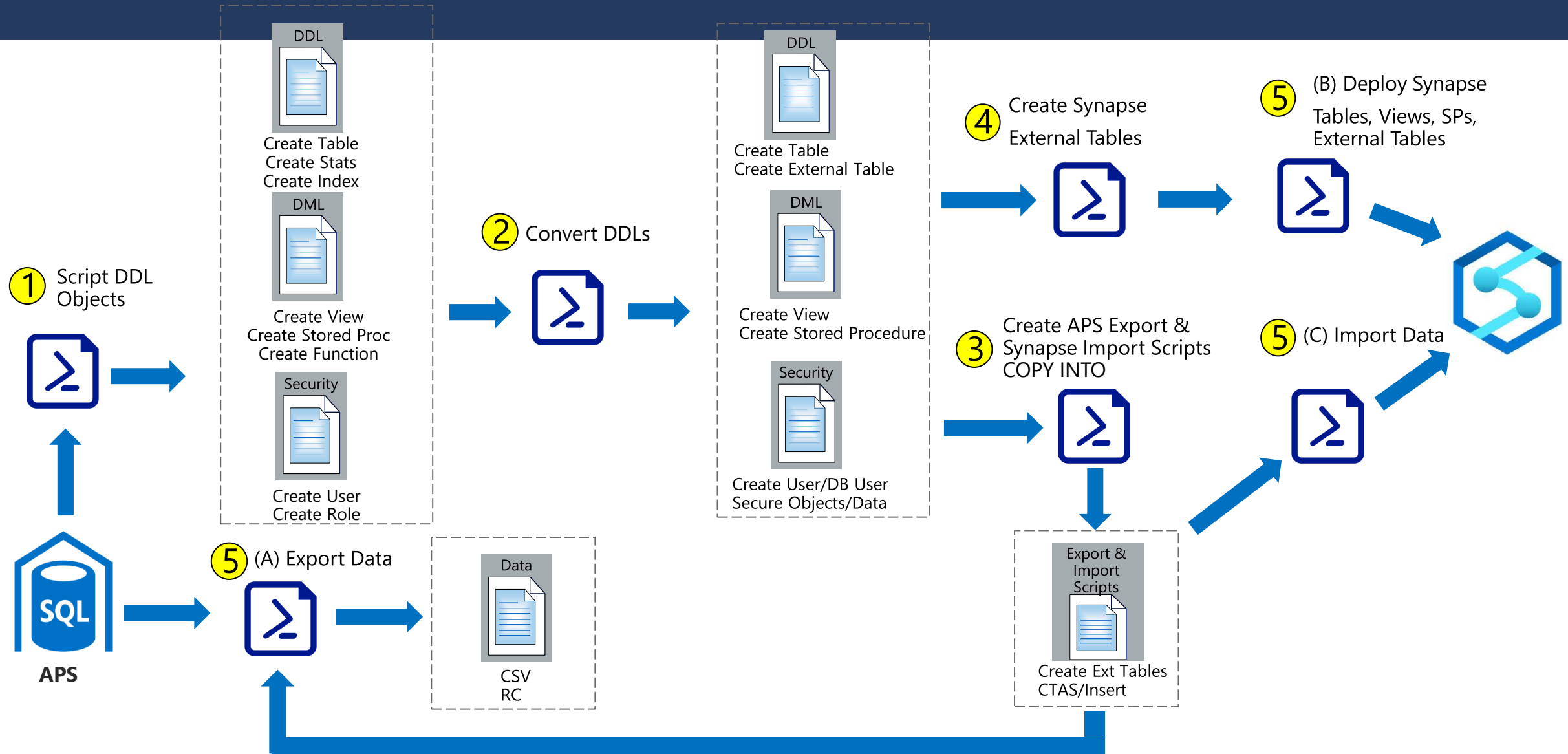
```
SET @getdate = (SELECT dbo.getdateDST(@getdate, @offset))
```

Migration Tools

Migration Tools Steps

1. Create DDL scripts from APS. **1_CreateDDLScripts**
2. Convert DDL scripts (Schema and #TEMP changes). **2_ConvertDDLScriptsDriver**
3. Create Data Export/Import Scripts. **3_CreateAPSExportScriptSynapseImportScript**
4. Create External Tables for Synapse. **4_CreateExternalTablesSynapse**
5. Run Scripts for Migration. **5_DeployScriptsToSynapse**
 - a) Export Data to Azure – Scripts from step 3 above
 - b) Create Tables/Views/SPs – Scripts from step 2 above (SP's require manual changes first)
 - c) Import Data to Synapse – Scripts from step 3 & 4 above

APS to Synapse Migration Process Flow

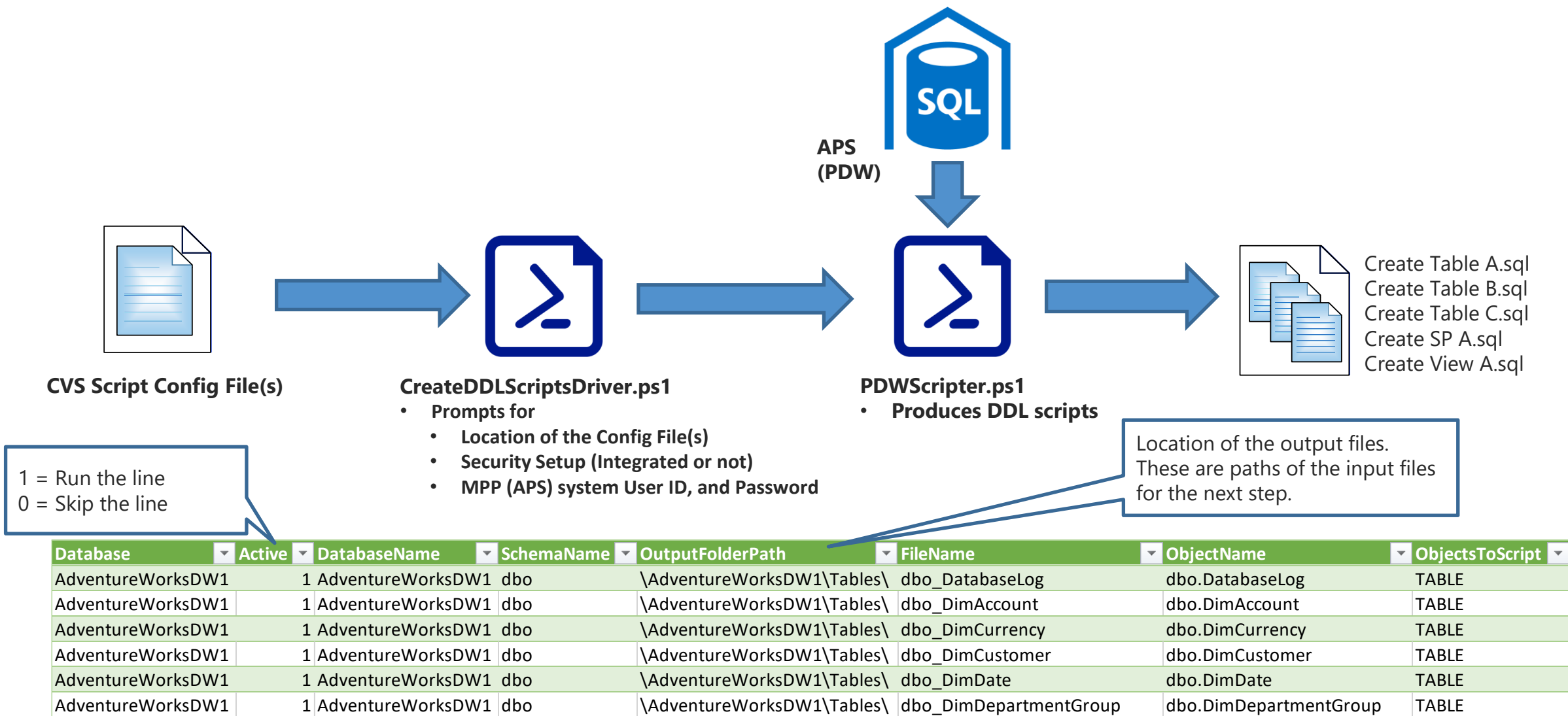


Migration Tools Steps – 1) Create DDL Scripts

1. Create config files for ScriptDDLScriptsDriver. This is a CSV file containing the following columns – see below.
2. Run CreateDDLScriptsDriver.ps1 and provide prompted info.

Parameter	Purpose	Value (Sample)
Database	Name of the DB to connect to	AdventureWorksDW
Active	1 – Run line, 0 – Skip line	0 or 1
DatabaseName	Name of the DB to connect to	AdventureWorksDW
SchemaName	Name of the object schema	dbo
OutputFolderPath	Relative path where the .sql file should be saved. Must have “\” on end.	\AdventureWorksDW\Tables\
FileName	Name of the file to store the script in. This should match the object name if possible	dbo_DimAccount
ObjectName	Name of the object to script. This must include the schema.	dbo.DimAccount
ObjectToScript	Used in logging only	SP, VIEW, TABLE, INDEX, STAT, ROLE, USER

Migration Tools Steps – 1) Create DDL Scripts



Migration Tools Steps – 2) Convert DDL scripts

All Schema changes are enforced by the mapping rules defined in a schema mapping config file. All changes are illustrated below, using the sample schema mapping file below.

Sample Schema Mapping (schemas.csv) :

ApsDbName	ApsSchema	SynapseSchema
AdventureWorksDW1	dbo	aw1
AdventureWorksDW2	dbo	aw2

#	Text	Replace with	Before (Example)	After (Example)
1	[APS_DB].[APS_Schema].[Table]	[Synapse_Schema].[Table]	CREATE/DELETE TABLE [AdventureWorksDW1].[dbo].[Tmp_Dates]	Create/Delete Table [aw1].[Tmp_Dates]
2	[APS_Schema].[Table]	[Synapse_Schema].[Table]	CREATE/DELETE TABLE [dbo].[Tmp_Dates]	CREATE/Delete TABLE [aw1].[Tmp_Dates]
3	[APS_DB].sys. Or APS_DB.sys.	.sys.	SELECT 1 FROM adventure_works.sys.schemas	SELECT 1 FROM sys.schemas
4	'[APS_DB_Name].[APS_Schema].[Table]'	'[Synapse_Schema].[Table]'	IF OBJECT_ID('[AdventureWorksDW1].[dbo].[DimAccount]') IF OBJECT_ID(N'[AdventureWorksDW2].[dbo].[DimAccount]')	IF OBJECT_ID('[aw1].[DimAccount]') IF OBJECT_ID(N'[aw2].[DimAccount]')
5	'[APS_Schema].[Table]'	'[Synapse_Schema].[Table]'	IF OBJECT_ID('[dbo].[DimAccount]') IF OBJECT_ID(N'[dbo].[DimAccount]')	IF OBJECT_ID('[aw1].[DimAccount]') If OBJECT_ID(N'[aw1].[DimAccount]')

Migration Tools Steps – 2) Convert DDL scripts

All DDL scripts are converted by applying the following changes:

- 1) If schema is omitted in object name, default schema is added to object name
- 2) Schema names are changed according to schema mapping
- 3) #TEMP tables syntax is adjusted. REPLICATE is replaced by ROUND_ROBIN. LOCATION=USER_DB is removed.

Adding omitted schema names

#	Text	Replace with	Before (Example)	After (Example)
1	FROM [Table]	FROM [Schema].[Table]	FROM [DimAccount]	FROM [dbo].[DimAccount]
2	EXECUTE StoredProcName	EXECUTE [Schema].[StoredProcName]	EXECUTE usp_CalculateExchangeRate	EXECUTE [dbo].[usp_CalculateExchangeRate]
3	CREATE VIEW ViewName	CREATE VIEW [Schema].ViewName	CREATE VIEW vExchangeRate	CREATE VIEW [dbo].vExchangeRate
4	CREATE TABLE Table	CREATE TABLE [Schema].Table	CREATE TABLE DimAccount	CREATE TABLE [dbo].DimAccount

#TEMP tables syntax adjustments

#	Text	Replace with
1	CREATE TABLE #TEMP WITH (DISTRIBUTION=REPLICATE)	CREATE TABLE #TEMP WITH (DISTRIBUTION=ROUND_ROBIN)
2	CREATE TABLE #TEMP WITH (DISTRIBUTION=REPLICATE, CLUSTERED COLUMNSTORE INDEX)	CREATE TABLE #TEMP WITH (DISTRIBUTION=ROUND_ROBIN, CLUSTERED COLUMNSTORE INDEX)
3	CREATE TABLE #TEMP WITH (HEAP, DISTRIBUTION=REPLICATE)	CREATE TABLE #TEMP WITH (HEAP, DISTRIBUTION=ROUND_ROBIN)
4	CREATE TABLE #TEMP WITH (HEAP, DISTRIBUTION=REPLICATE, LOCATION=USER_DB)	CREATE TABLE #TEMP WITH (HEAP, DISTRIBUTION=ROUND_ROBIN)

Migration Tools Steps – 2) Convert DDL scripts

1. Create schema mapping configuration file. This is a CSV file containing the following columns:

Parameter	Purpose	Value (Sample)
ApsDbName	Name of APS Database	AdventureWorksDW
ApsSchema	Name of the schema in APS database	dbo
SynapseSchema	Name of the schema in Synapse database	aw1

2. Create input and output directory configuration file. This is a CSV file containing the following columns:

Parameter	Purpose	Value (Sample)
Active	1 – Run line, 0 – Skip line	0 or 1
ApsDatabasesName	The name of APS database	AdventureWorksDW
SourceDirectory	Directory where the input source files that has APS schema names. This is the output files from previous step.	C:\AzureSynapseScriptsAndAccelerators\Migration\APS\Output\1_CreateMPPScripts\AdventureWorksDW1\Tables
TargetDirectory	Output directory of this step, where the scripts with new Synapse schemas will reside	C:\AzureSynapseScriptsAndAccelerators\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables
DefaultSchema	The name of default schema for this database	dbo
ObjectType	Type of the object	Table, View, SP, Index, Statistic, Function, Role, User

3. Run PowerShell script **ConvertDDLScriptsDriver.ps1** with prompted info.

Migration Tools Steps – 2) Convert DDL scripts

The schema mapping configuration file needs to be created manually. The default name is **schemas.csv**. It is a simple step to create the CSV file by first creating contents in an excel file (.xlsx), and then save it to a CSV file (.csv).

Manually create this Sample schema mapping .xlsx file:
File Name: **schemas_sample.xlsx**

ApsDbName	ApsSchema	SynapseSchema
CS_STAGE	ADM	ST_ADM
CS_STAGE	BAS	ST_BAS
CS_STAGE	CSS	ST_CSS
CS_STAGE	HR	ST_HR
CS_STAGE	SALES	ST_SALES
CS_WAREHOUSE	ADM	DW_ADM
CS_WAREHOUSE	BAS	DW_BAS
CS_WAREHOUSE	CSS	DW_CSS
CS_WAREHOUSE	HR	DW_HR
CS_WAREHOUSE	SALES	DW_SALES
CS_DATAMART	ADM	DM_ADM
CS_DATAMART	BAS	DM_BAS
CS_DATAMART	CSS	DM_CSS
CS_DATAMART	HR	DM_HR
CS_DATAMART	SALES	DM_SALES

Save As CSV File
"schemas_sample.csv"

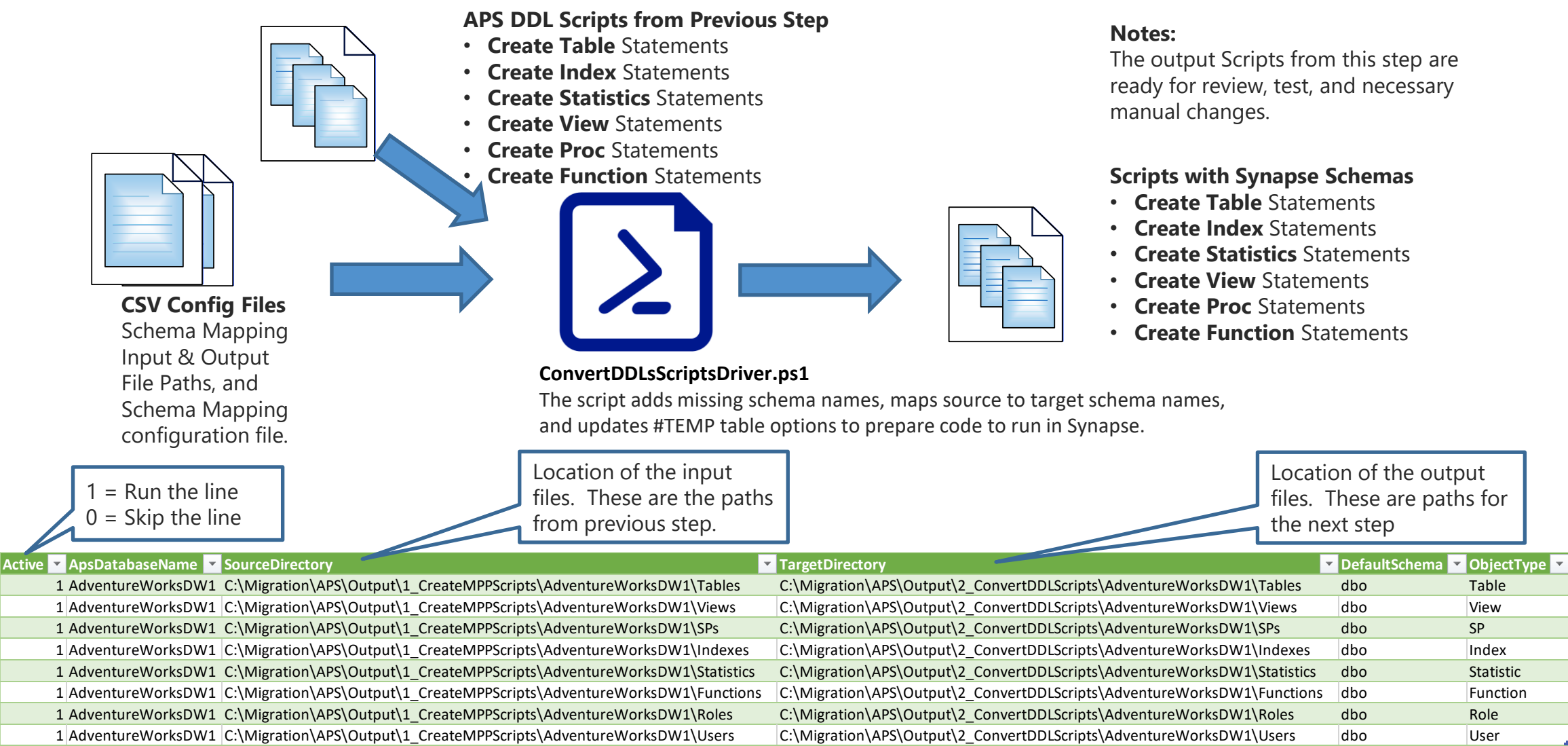


Schema **CSS** in all 3 APS DBs.
It is mapped to different names
in Synapse: **ST_CSS, DW_CSS,
DM_CSS**

As Input to ConvertDDLScriptsDriver.ps1 script.
File Name: **schemas_sample.csv**

ApsDbName	ApsSchema	SynapseSchema
CS_STAGE	ADM	ST_ADM
CS_STAGE	BAS	ST_BAS
CS_STAGE	CSS	ST_CSS
CS_STAGE	HR	ST_HR
CS_STAGE	SALES	ST_SALES
CS_WAREHOUSE	ADM	DW_ADM
CS_WAREHOUSE	BAS	DW_BAS
CS_WAREHOUSE	CSS	DW_CSS
CS_WAREHOUSE	HR	DW_HR
CS_WAREHOUSE	SALES	DW_SALES
CS_DATAMART	ADM	DM_ADM
CS_DATAMART	BAS	DM_BAS
CS_DATAMART	CSS	DM_CSS
CS_DATAMART	HR	DM_HR
CS_DATAMART	SALES	DM_SALES

Migration Tools Steps – 2) Convert DDL scripts



Migration Tools Steps – 3) CETAS(APS)/CTAS(Synapse)

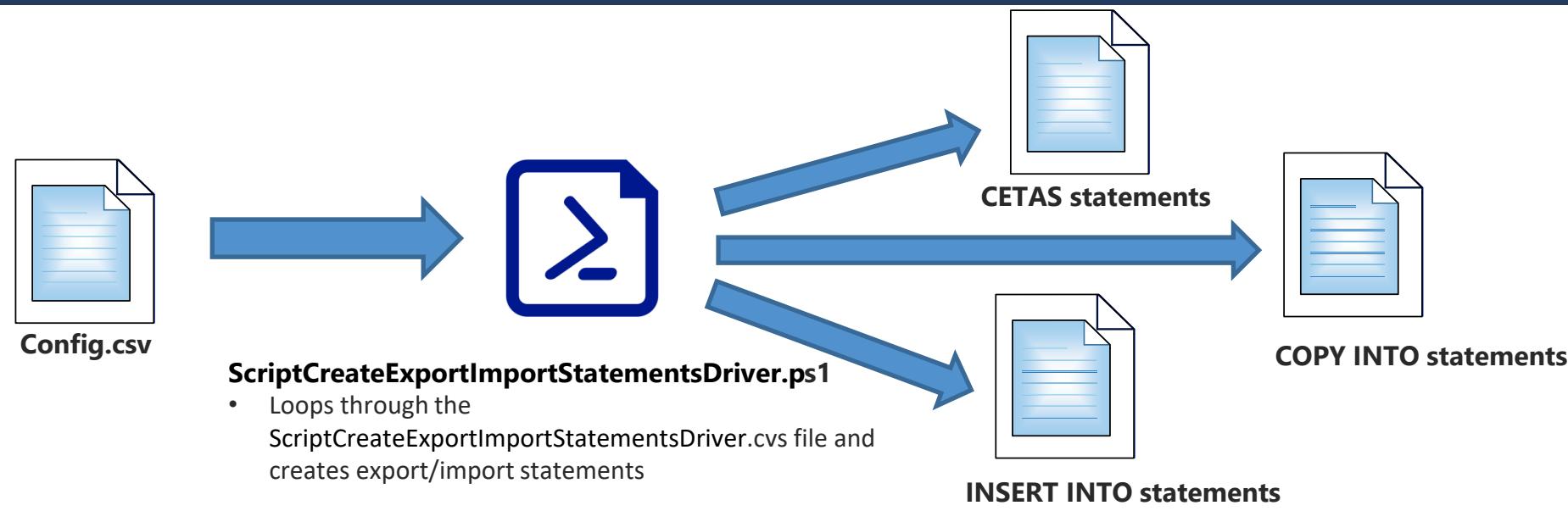
Exporting data from APS to Synapse can be accomplished using Polybase. This script generates the export and import statements necessary to move the data.

1. Create a config file in the format below – **see the next slide.**
2. Run the script ScriptCreateExportImportStatementDriver.ps1. The script does not connect to the APS or Synapse. The only input for this script is the config.csv file

Migration Tools Steps – 3) CETAS(APS)/CTAS(Synapse)

Parameter	Purpose	Value (Sample)
Active	1 – Run line, 0 – Skip line	0 or 1
DatabaseName	Name of the database in APS	AdventureWorksDW
OutputFolderPath	Name of the path to output the results to	C:\...\Migration\APS\Output\3_CreateAPSExportScriptSynapseImportScript\ExportAPS\AdventureWorksDW1
FileName	Name of the output file	DimAccount
SourceSchemaName	Name of the APS/Source Schema	Dbo
SourceObjectName	Name of the source object to work with	DimAccount
DestSchemaName	Name of the destination schema in Synapse	dbo
DestObjectName	Name of the destination object	DimAccount
DataSource	Name of the data source to use. This must already be created.	AZURE_BLOB_STORAGE
FileFormat	Name of the File Format to use when exporting the data. This must already be created.	DelimitedNoDateZip
ExportLocation	Folder path in the staging container. Each Table should have its own file location	/AdventureWorksDW1/dbo_DimAccount
InsertFilePath	Path to write the import statements	C:\...\Migration\APS\Output\3_CreateAPSExportScriptSynapseImportScript\ImportSynapse\AdventureWorksDW1\
CopyFilePath	Path to write the COPY statements	C:\...\Migration\APS\Output\3_CreateAPSExportScriptSynapseImportScript\CopySynapse\AdventureWorksDW1\
ImportSchema	Name of the new schema in Synapse	aw1
StorageAccountName	Name of the staging storage account	Apsmigrationstaging
ContainerName	Name of the container in staging storage account	aps-Polybase

Migration Tools Steps – 3) CETAS(APS)/CTAS(Synapse)



Active	DatabaseName	OutputFolderPath	FileName	SourceSchemaName	SourceObjectName	DestSchemaName
1	AdventureWorksDW1	C:\...\3_CreateAPSExportScriptSynapseImportScript\ExportAPS\...\	dbo_DatabaseLog	dbo	DatabaseLog	EXT_aw1
1	AdventureWorksDW1	C:\...\3_CreateAPSExportScriptSynapseImportScript\ExportAPS\...\	dbo_DimAccount	dbo	DimAccount	EXT_aw1
1	AdventureWorksDW1	C:\...\3_CreateAPSExportScriptSynapseImportScript\ExportAPS\...\	dbo_DimCurrency	dbo	DimCurrency	EXT_aw1
1	AdventureWorksDW1	C:\...\3_CreateAPSExportScriptSynapseImportScript\ExportAPS\...\	dbo_DimCustomer	dbo	DimCustomer	EXT_aw1
1	AdventureWorksDW1	C:\...\3_CreateAPSExportScriptSynapseImportScript\ExportAPS\...\	dbo_DimDate	dbo	DimDate	EXT_aw1

DestObjectName	DataSource	FileFormat	ExportLocation	InsertFilePath	CopyFilePath	ImportSchema	StorageAccountName	ContainerName
DatabaseLog	AZURE_BLOB_STORAGE	DelimitedNoDateZIP	/AdventureWorksDW1/dbo_DatabaseLog	C:\...\ImportSynapse\...\	C:\...\CopySynapse\...\	aw1	apsmigrationstaging	aps-polybase
DimAccount	AZURE_BLOB_STORAGE	DelimitedNoDateZIP	/AdventureWorksDW1/dbo_DimAccount	C:\...\ImportSynapse\...\	C:\...\CopySynapse\...\	aw1	apsmigrationstaging	aps-polybase
DimCurrency	AZURE_BLOB_STORAGE	DelimitedNoDateZIP	/AdventureWorksDW1/dbo_DimCurrency	C:\...\ImportSynapse\...\	C:\...\CopySynapse\...\	aw1	apsmigrationstaging	aps-polybase
DimCustomer	AZURE_BLOB_STORAGE	DelimitedNoDateZIP	/AdventureWorksDW1/dbo_DimCustomer	C:\...\ImportSynapse\...\	C:\...\CopySynapse\...\	aw1	apsmigrationstaging	aps-polybase
DimDate	AZURE_BLOB_STORAGE	DelimitedNoDateZIP	/AdventureWorksDW1/dbo_DimDate	C:\...\ImportSynapse\...\	C:\...\CopySynapse\...\	aw1	apsmigrationstaging	aps-polybase

Migration Tools Steps – 4) Create ext. tables Synapse

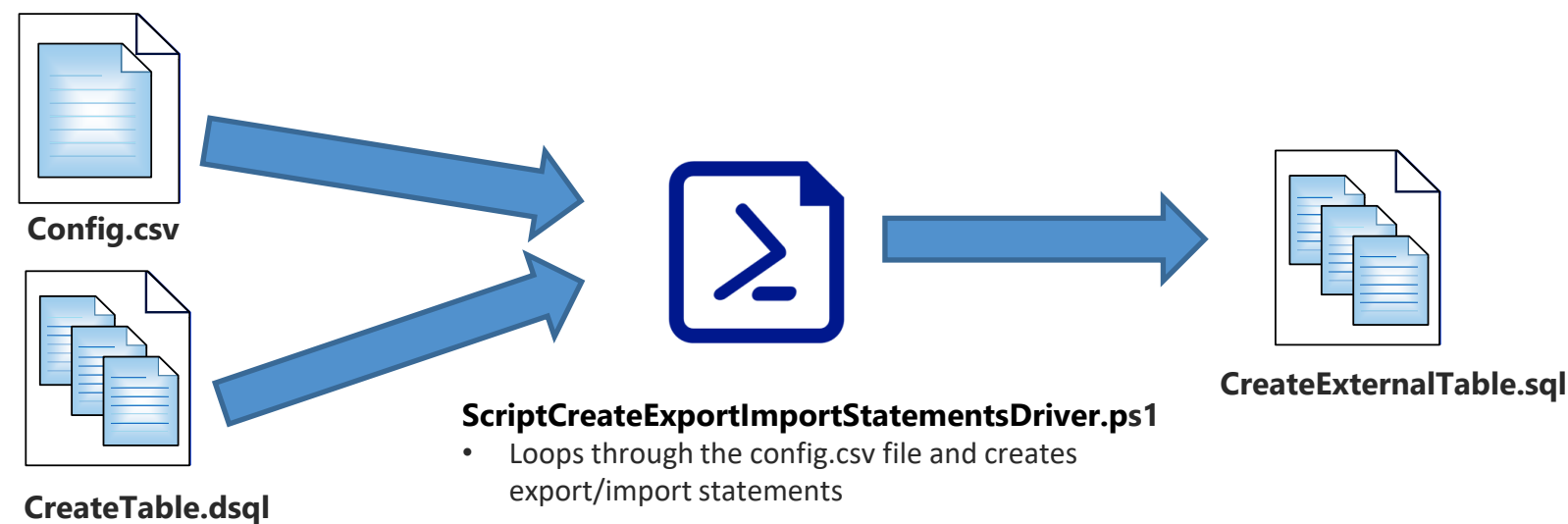
After the data has been exported from APS, the data now needs to be inserted into Synapse. Before this can occur, the external table needs to be created in Synapse. This is completed by using the create table statements and converting the statement into an external table.

1. Create a config.csv file in the format below – **see the next slide.**
2. Run the script ScriptCreateExternalTableDriver_V1.ps1. The script does not connect to the APS or Synapse. The only input for this script is the config.csv file

Migration Tools Steps – 4) Create ext. tables Synapse

Parameter	Purpose	Value (Sample)
Active	1 – Run line, 0 – Skip line	0 or 1
OutputFolderPath	Name of the path where output files will be stored	C:\AzureSynapseScriptsAndAccelerators\Migration\APS\Output\4_CreateExternalTablesSynapse\AdventureWorksDW1\
FileName	Name of the output file	DimCusDimAccounttomer
InputFolderPath	Path to the create Table output from step 2	C:\AzureSynapseScriptsAndAccelerators\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables\
InputFileName	Name of the Create Table script	DimAccount.dsql
SchemaName	Name of the schema to create the external table in	dbo
ObjectName	Name of the external table to create	ext_DimAccount
DateSource	Name of the data source to use for the external table	AZURE_BLOB_STORAGE
FileFormat	Name of the File Format to use when exporting the data. Must already be created.	DelimitedNoDateZip
FileLocation	Folder path in the staging container. Each Table should have its own file location.	/AdventureWorksDW1/dbo_DimAccount

Migration Tools Steps – 4) Create ext. tables Synapse



Active	OutputFolderPath	FileName	InputFolderPath
1	C:\Migration\APS\Output\4_CreateExternalTablesSynapse\AdventureWorksDW1\	EXT_aw1_DatabaseLog	C:\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables\
1	C:\Migration\APS\Output\4_CreateExternalTablesSynapse\AdventureWorksDW1\	EXT_aw1_DimAccount	C:\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables\
1	C:\Migration\APS\Output\4_CreateExternalTablesSynapse\AdventureWorksDW1\	EXT_aw1_DimCurrency	C:\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables\
1	C:\Migration\APS\Output\4_CreateExternalTablesSynapse\AdventureWorksDW1\	EXT_aw1_DimCustomer	C:\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables\
1	C:\Migration\APS\Output\4_CreateExternalTablesSynapse\AdventureWorksDW1\	EXT_aw1_DimDate	C:\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables\

InputFileName	SchemaName	ObjectName	DataSource	FileFormat	FileLocation
dbo_DatabaseLog.dsql	EXT_aw1	DatabaseLog	AZURE_BLOB_STORAGE	DelimitedNoDateZIP	/AdventureWorksDW1/dbo_DatabaseLog
dbo_DimAccount.dsql	EXT_aw1	DimAccount	AZURE_BLOB_STORAGE	DelimitedNoDateZIP	/AdventureWorksDW1/dbo_DimAccount
dbo_DimCurrency.dsql	EXT_aw1	DimCurrency	AZURE_BLOB_STORAGE	DelimitedNoDateZIP	/AdventureWorksDW1/dbo_DimCurrency
dbo_DimCustomer.dsql	EXT_aw1	DimCustomer	AZURE_BLOB_STORAGE	DelimitedNoDateZIP	/AdventureWorksDW1/dbo_DimCustomer
dbo_DimDate.dsql	EXT_aw1	DimDate	AZURE_BLOB_STORAGE	DelimitedNoDateZIP	/AdventureWorksDW1/dbo_DimDate

Migration Tools Steps – 5) Deploy Scripts

The Deployment script is designed to run any .sql file. For the purpose of the migration, it can be used to deploy objects to Synapse. This tool can drop an existing object before running the .sql file.

1. Copy the Scripts from Source Repository and Place them on in a local directory.
 - ❑ Any directory structure will work. As a suggestion this path can be used:
C:\AzureSynapseScriptsAndAccelerators\Migration\APS\5_DeployScriptsToSynapse.
 - ❑ Place the two PowerShell scripts in the above directory ([RunDSQLScriptDriver.ps1](#) and [RunSQLScriptFile.ps1](#))
 - ❑ You can choose to put all your CSV configuration files under the above directory, or in a separate directory under it, such as:
C:\AzureSynapseScriptsAndAccelerators\Migration\APS\5_DeployScriptsToSynapse\Config_Files
2. Select one of the sample configuration files for the purpose of your deployment. All the three sample configuration files use the same format. Sample configuration files provided:
 - ❑ Export APS Data to Azure Blob Storage: [ApsCreateExtTablesAndExportData.csv](#)
 - ❑ Create Tables/Views/SPs in Azure Synapse: [SynapseCreateTablesViewsAndSPs.csv](#)
 - ❑ Import APS Data to Azure Blob Storage: [SynapseImportData.csv](#)

Migration Tools Steps – 5) Deploy Scripts

Edit the one of the sample config files to fit the purpose of your deployment, referring to the definition of each column below:

Parameter	Purpose	Value (Sample)
Active	1 – Run line, 0 – Skip line	0 or 1
ServerName	Name of the SQL Server/Synapse/APS(PDW)	synapseserver.database.windows.net
DatabaseName	Name of the DB to connect to	AdventureWorksDW
FilePath	Path to the script that needs to be run. Do not put a '\' on the end.	C:\AzureSynapseScriptsAndAccelerators\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables
CreateSchema	1 – Create Schema, 0 – Don't create Schema	0 or 1
ObjectType	Type of object to Create. Used to create the drop statement. Valid Values: "", TABLE, VIEW, SP, (SCHEMA, STAT – Not implemented yet)	TABLE, VIEW, SP, SCHEMA, EXT
SchemaAuth	Should a Schema Authorization be needed when creating the schema, enter the name of the Authorization to use. If left empty, no authorization is created.	Login to Create Schema
SchemaName	Schema Name for the object to be created.	aw1
ObjectName	Name of the object that is being created. Used in creating the drop statement and logging.	Name of the object
ParentObjectName	Name of the parent object. Valid for statistics and indexes.	DimAccount
DropIfExists	DROP – Drop object if already exists, TRUNCATE - Truncate Table if exists, NO – Do not drop or Truncate if exist.	DROP
FileName	The name of the script file	dbo_DimAccount.dsql

Migration Tools Steps – 5) Deploy Scripts

4. Run the PowerShell script (RunDSQLScriptsDriver.ps1). This script will prompt for the following information

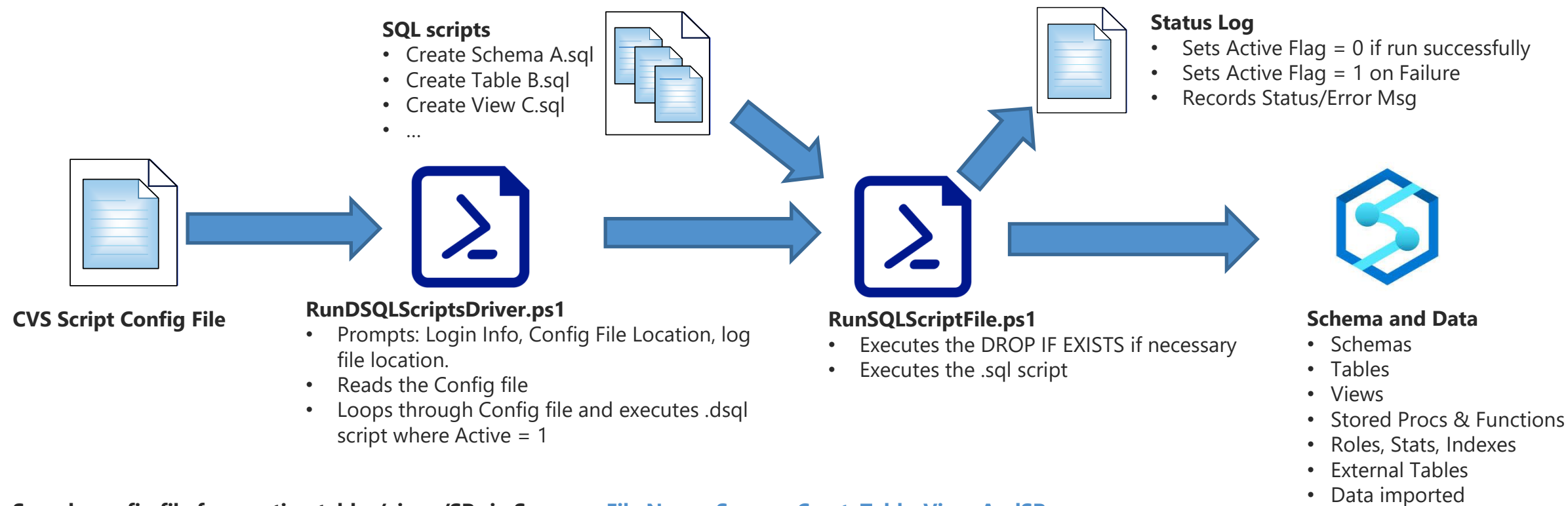
- “Enter the name of the Script Config csv File.” – This will be the location\name of your configuration file.
 - C:\AzureSynapseScriptsAndAccelerators\Migration\APS\5_DeployScriptsToSynapse: [SynapseCreateTablesViewsAndSPs.csv](#) or [ApsCreateExtTablesAndExportData.csv](#) or [SynapseImportData.csv](#)
- “How do you want to connect to SQL(ADPass, ADInt, WinInt, SQLAuth)?”
 - ADPass – This should be used for SQL Authentication with Password (Azure)
 - ADINT – Azure AD Authentication
 - SQLAUTH – SQL Server Authentication with username and password.
 - “Blank” – AD integrated Authentication
- “Enter the User Name to Connect to the SQL Server.” – User name with permission to create objects
- “Enter the Password for the User” – Enter the Password for the user – reads password as a secure string
- “Enter the name of the Output File Directory.” – Enter the location where the output log will be written
- “Enter the name of the status file.” – Enter the name of the Status File

5. Review the Status log for Success Failures.

Review the status log file. The file name and location are the prompted values of the PowerShell program in step 3). The default location is the location of the PowerShell scripts with the file name [status.csv](#).

- Should a failure occur, the Status log will set the Active flag to 0 for all successful objects created. The Failures will remain Active = 1. This will allow the status log to be used as the Script Config file and only the failed objects will be run

Migration Tools Steps – 5) Deploy Scripts



Sample config file for creating tables/views/SPs in Synapse. [File Name: SynapseCreateTablesViewsAndSPs.csv](#)

Active	ServerName	DatabaseName	FilePath	CreateSchema	ObjectType	SchemaAuth	SchemaName	ObjectName	ParentObjectName	DropTruncatelfExists	Variables	FileName
1	synapseserver.database.windows.net	synapse	C:\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables	1	Table	dbo	aw1	DatabaseLog		DROP		dbo_DatabaseLog.dsql
1	synapseserver.database.windows.net	synapse	C:\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables	1	Table	dbo	aw1	DimAccount		DROP		dbo_DimAccount.dsql
1	synapseserver.database.windows.net	synapse	C:\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables	1	Table	dbo	aw1	DimCurrency		DROP		dbo_DimCurrency.dsql
1	synapseserver.database.windows.net	synapse	C:\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables	1	Table	dbo	aw1	DimCustomer		DROP		dbo_DimCustomer.dsql
1	synapseserver.database.windows.net	synapse	C:\Migration\APS\Output\2_ConvertDDLScripts\AdventureWorksDW1\Tables	1	Table	dbo	aw1	DimDate		DROP		dbo_DimDate.dsql

Change Examples

Table Schema Change

The schema mapping must be performed for all "Create Table" statements. 'Heap' Table must be specified explicitly in Synapse. Otherwise, Synapse creates Clustered ColumnStore Index (CCI) for the table.

```
-- APS
CREATE TABLE [dbo].[DimAccount]
(
...
)
WITH(CLUSTERED INDEX ([AccountKey] ASC),
DISTRIBUTION = REPLICATE)
```

```
-- Synapse
CREATE TABLE [aw1].[DimAccount]
(
...
)
WITH(CLUSTERED INDEX ([AccountKey] ASC),
DISTRIBUTION = REPLICATE)
```

Stats and Index Change

To speed up the data migration, we will remove "Create Index" and "Create Statistics" statements, put them in a separate file, update the schemas, save the file for later use. After the data migration, these Indexes and Statistics will be created in Synapse environment.

```
-- APS
CREATE INDEX [idx_ProductKey]
ON [dbo].[DimProduct]
([ProductKey])

CREATE STATISTICS [stat_ProductColor]
ON [dbo].[DimProduct]
([Color])
```

```
-- Synapse
CREATE INDEX [idx_ProductKey]
ON [aw1].[DimProduct]
([ProductKey])

CREATE STATISTICS [stat_ProductColor]
ON [aw1].[DimProduct]
([Color])
```

View Schema Change

Schema mapping in Views. Same DDL schema mapping rules apply to Views.

```
-- APS  
CREATE VIEW [dbo].[vDimProduct] AS  
SELECT *  
FROM dbo.DimProduct  
WHERE Color='Black';
```

```
-- Synapse  
CREATE VIEW [aw1].[vDimProduct] AS  
SELECT *  
FROM dbo.DimProduct  
WHERE Color='Black';
```

Stored Procedure Schema Change

The new schemas of the target Synapse Stored Procedures will replace APS schemas, with the mappings defined in the schema mapping configuration file ('schema.cs') explained earlier.

```
-- APS
CREATE PROC [dbo].[sp_GetBlackProducts]
AS
BEGIN
    CREATE TABLE dbo.DimProduct
    WITH (DISTRIBUTION = HASH
    ([Product_Key]), HEAP)
    AS
    SELECT *
    FROM [dbo].DimProduct WHERE
    Color='Black'
END
```

```
-- Synapse
CREATE PROC [aw1].[sp_GetBlackProducts]
AS
BEGIN
    CREATE TABLE aw1.DimProduct
    WITH (DISTRIBUTION = HASH
    ([Product_Key]), HEAP)
    AS
    SELECT *
    FROM [aw1].DimProduct WHERE
    Color='Black'
END
```

Q&A



Thank you