Build large-scale data loading pipelines with metadata-driven approach in copy data tool

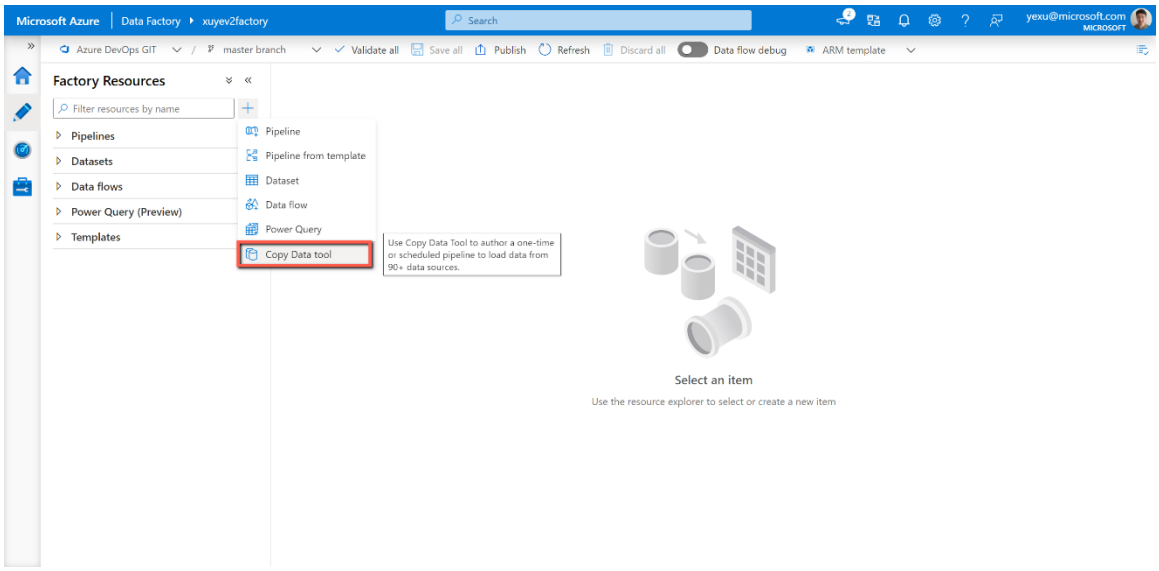
When you want to copy huge amounts of objects (e.g. thousands of tables) or load data from big variety of sources, the appropriate approach is to input the name list of the objects and required copy behaviors to a control table, and then use parameterized pipelines to read the same from the control table and apply them to the downstream jobs accordingly. By doing so, you can maintain (add/remove) the objects list to be copied easily by just updating the object names in control table instead of redeploying the pipelines. What's more, you will have single place to easily check which objects copied by which pipelines/triggers with what kind of copy behaviors.

ADF copy data tool ease the journey of building such kind of metadata driven data loading pipelines. After you go through an intuitive flow from a wizard based expereince, the tool can generate parameterized pipelines and SQL scripts for you to create external control tables accordingly. After you run the generated scripts to create the control table in your SQL database, your pipelines will read the metadata from the control table and apply them on the copy jobs automatically.

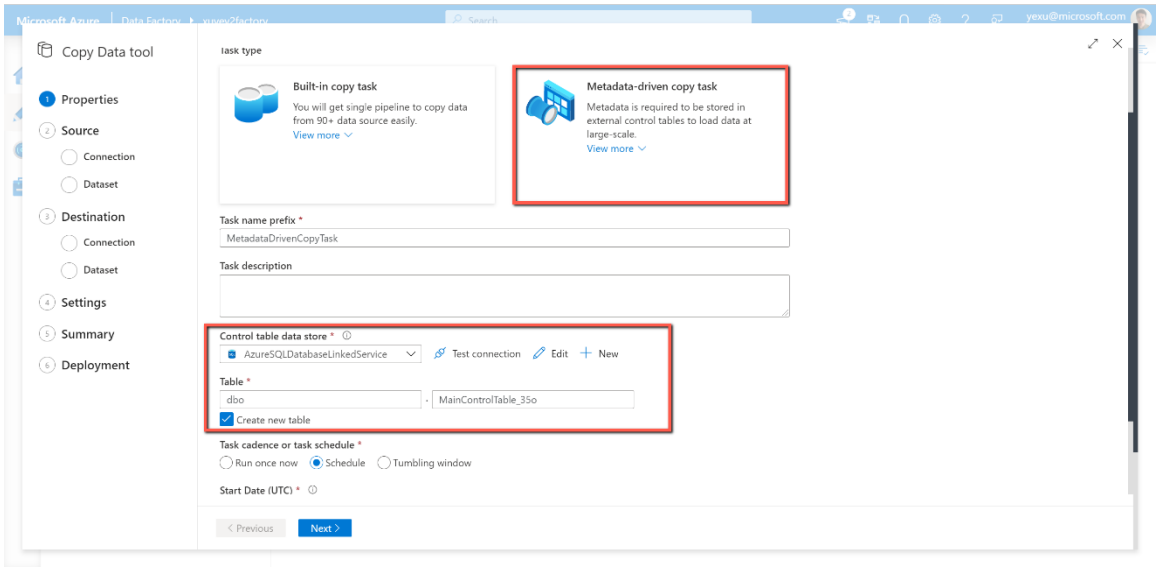
How to get start:

1. Add feature flag to be whitelisted on this feature: https://adf.azure.com?feature.enableMetadataDrivenSolution=true

2. Open copy data tool

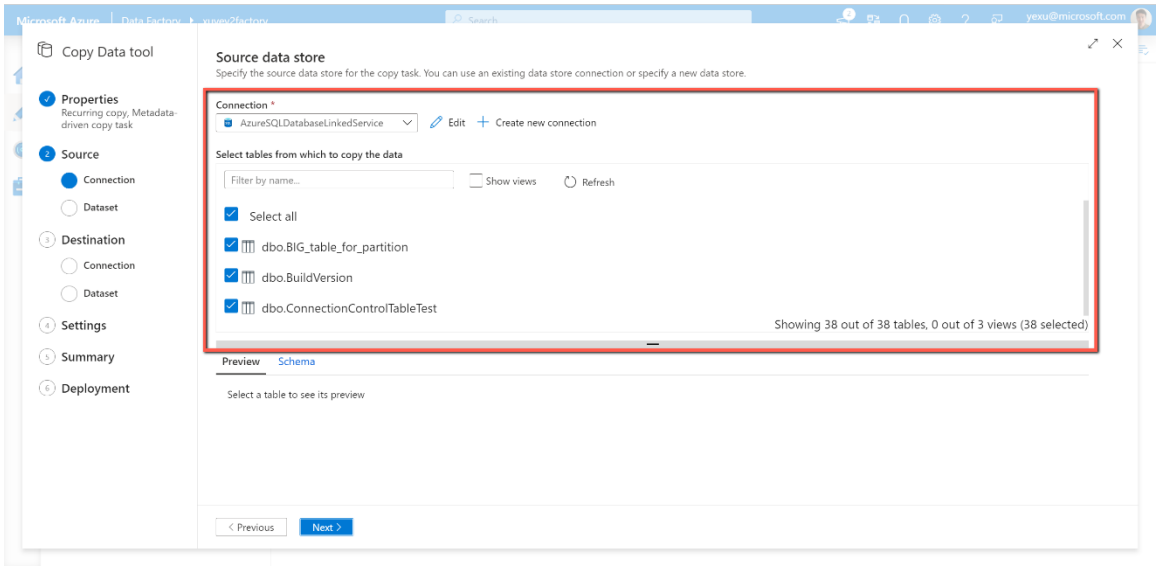


3. Select **Metadata-driven copy task**



You need to select the **connection to your control table** and input the **control table name**. The generated pipeline via copy data tool will read metadata from that particular control table.

4. Input the **connection of your source database** and then select the **source table name** you want to copy.

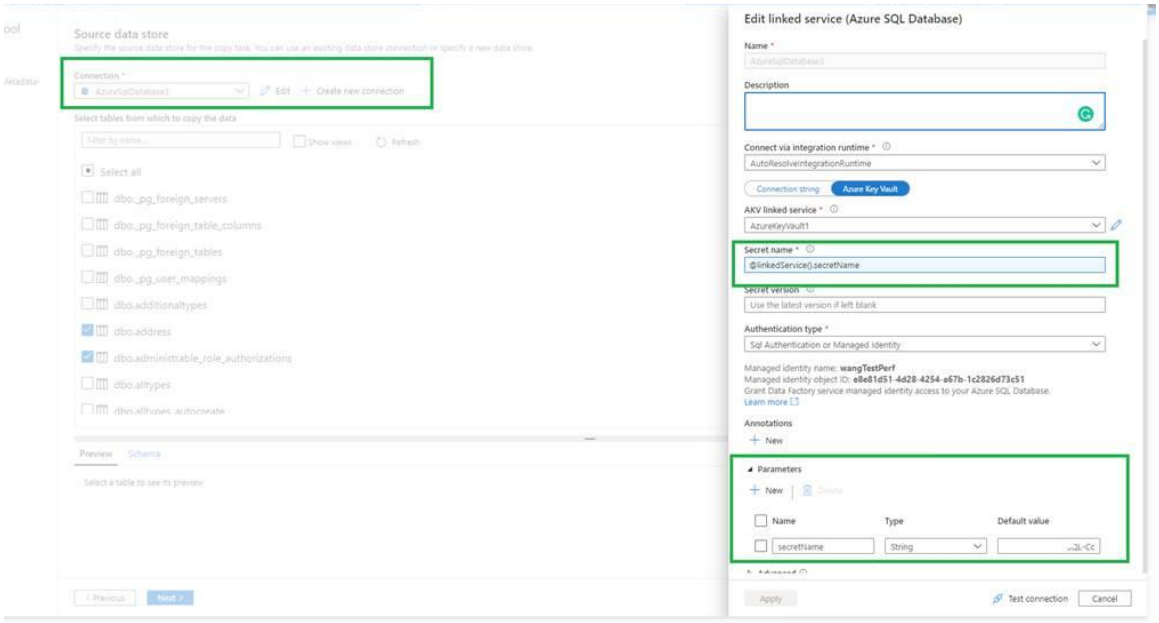


Please note:

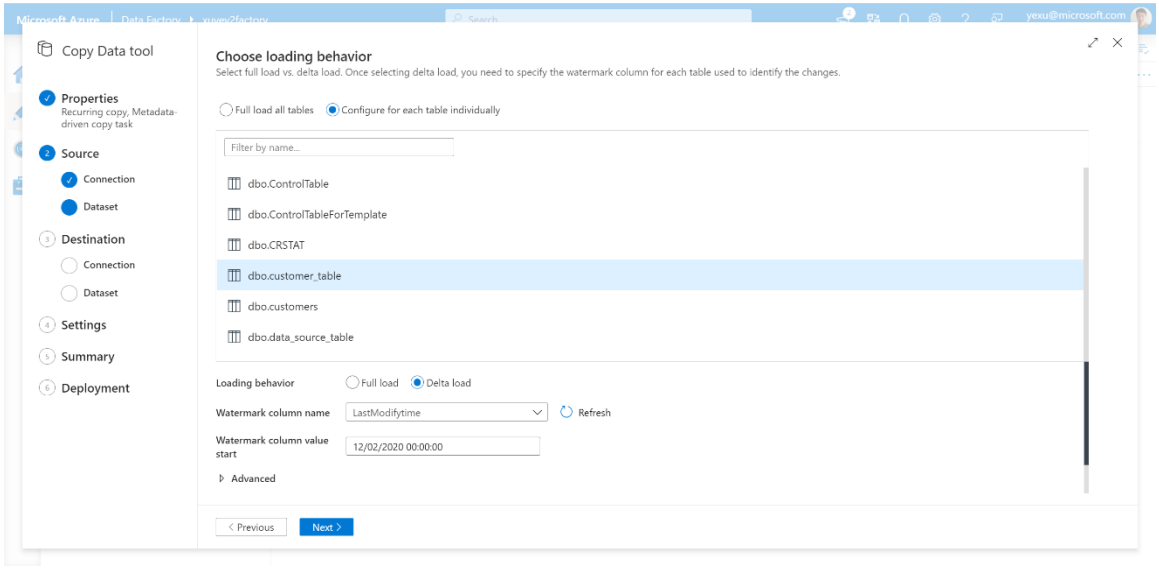
- If you select tabular data store on this page, you can further configure either full load or incremental load in the next page.
- If you select storage store in this page, you can do full load only in the next page. (Incremental load from storage store is not supported today)

5. (Optional) You can also use parameterized linked service with real connection string value written in control table (table name: connection control table). By doing so, this generated pipeline can be used to copy data from multiple DBs or servers to the destination.

   More details on [parameterized linked service]



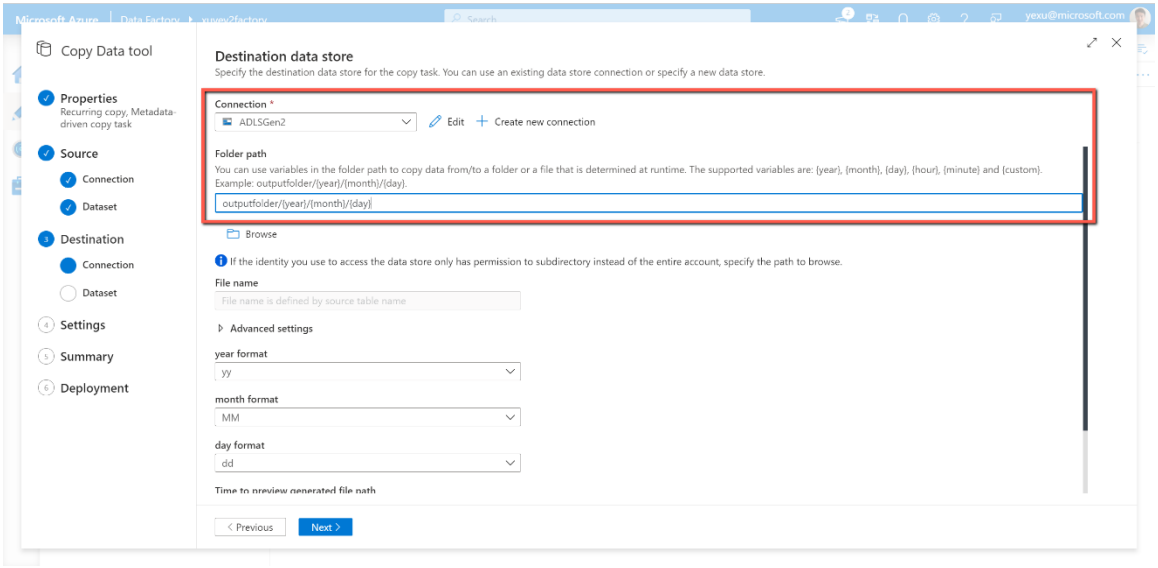
6. Select the **loading behavior**



If you want to do full load on all the tables, select **Full load all tables.**

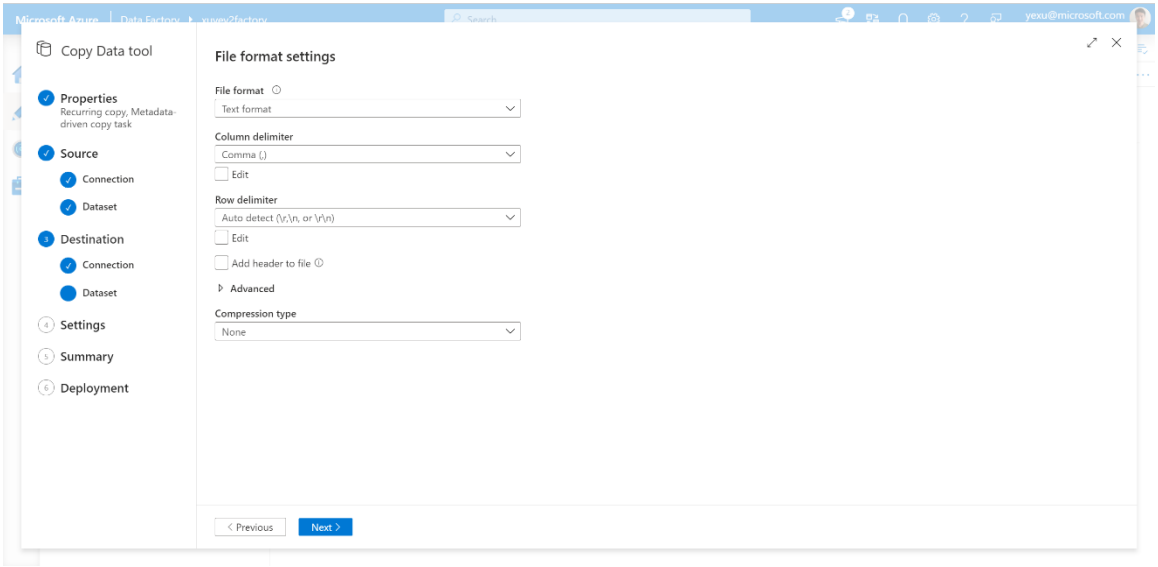
If you want to do incremental load, you can select **configure for each table individually**, and select **Delta load** as well as **watermark column & value to start** for each table.

7. Input the **connection of your destination store** and the **folder path** you want to copy data to

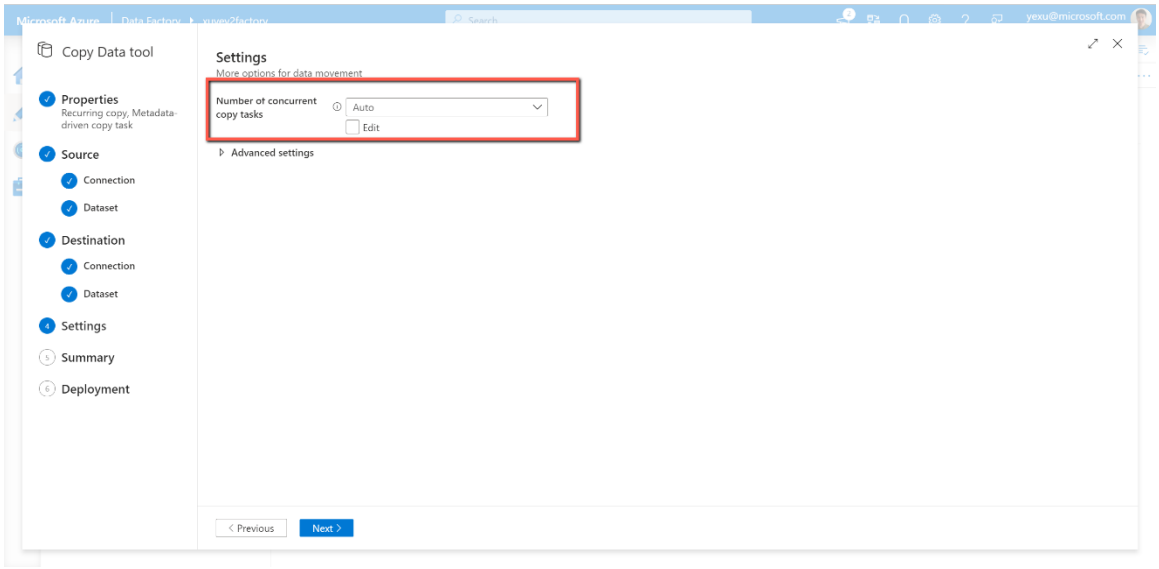


You can input dynamic value in folder path like “outputfolder/{year}/{month}/{day}”. If you select the dynamic value, the destination folder path will not be written into the control table but part of the generated pipeline.

8. Input file **format settings**

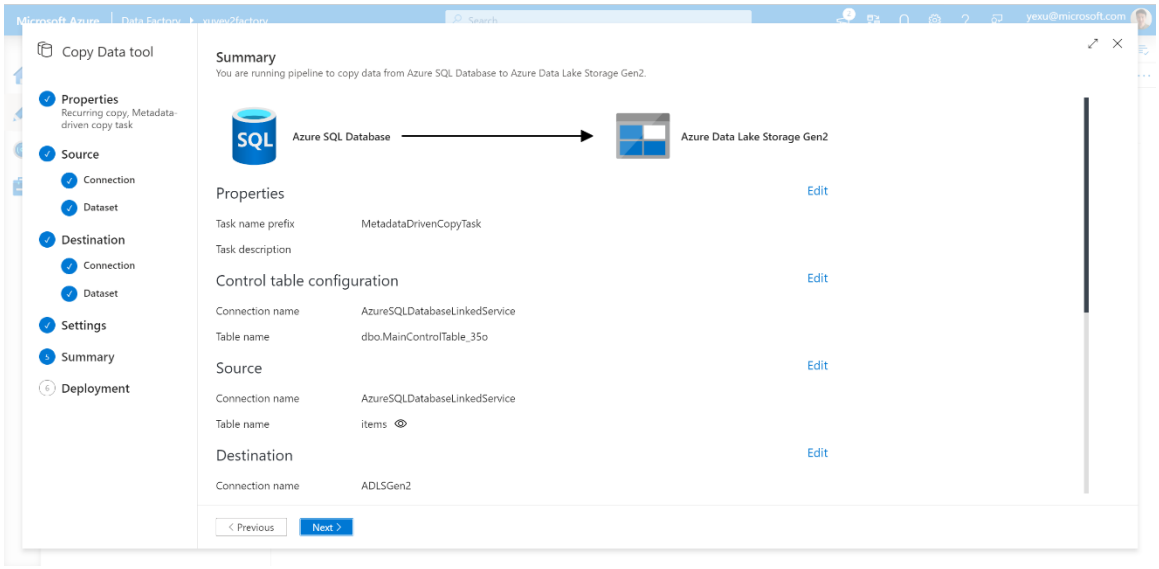


9. Select the **number of concurrent copy task**

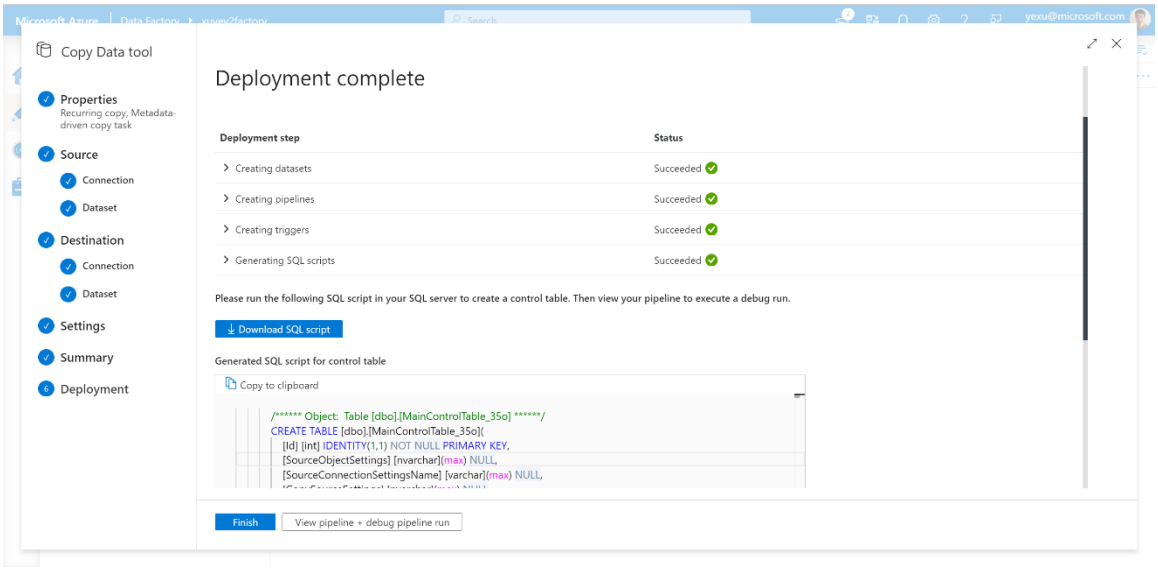


You can decide the max number of concurrent copy activity run in order to control the load impact on your source store. The default value is 20. It means by default 20 copy activity runs in the generated pipeline will load data in parallel from your source database.

10. Validate the **summary**



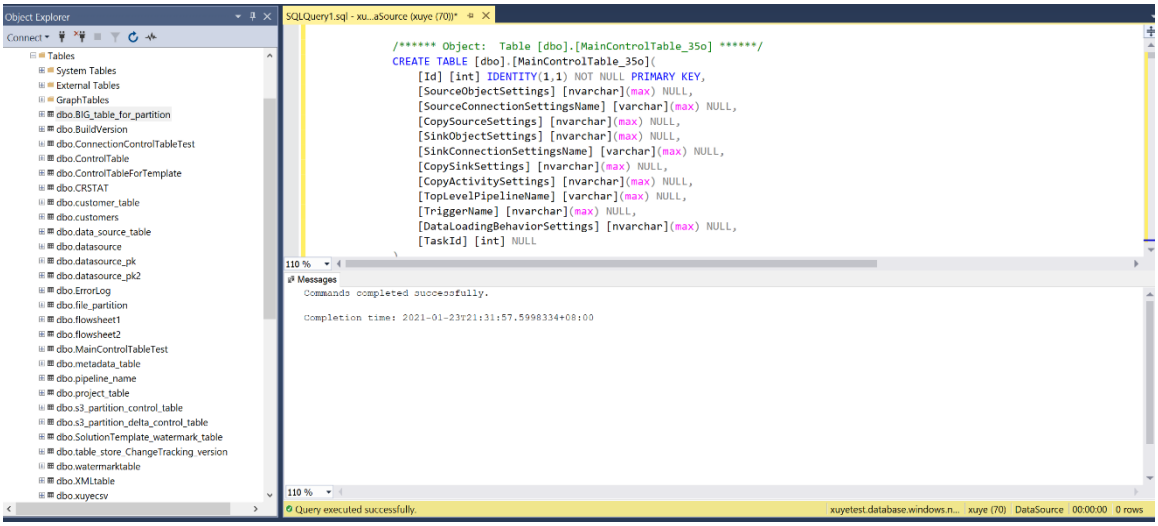
11. Copy or download the **SQL scripts for control table** from UI to create your control table and store procedure.



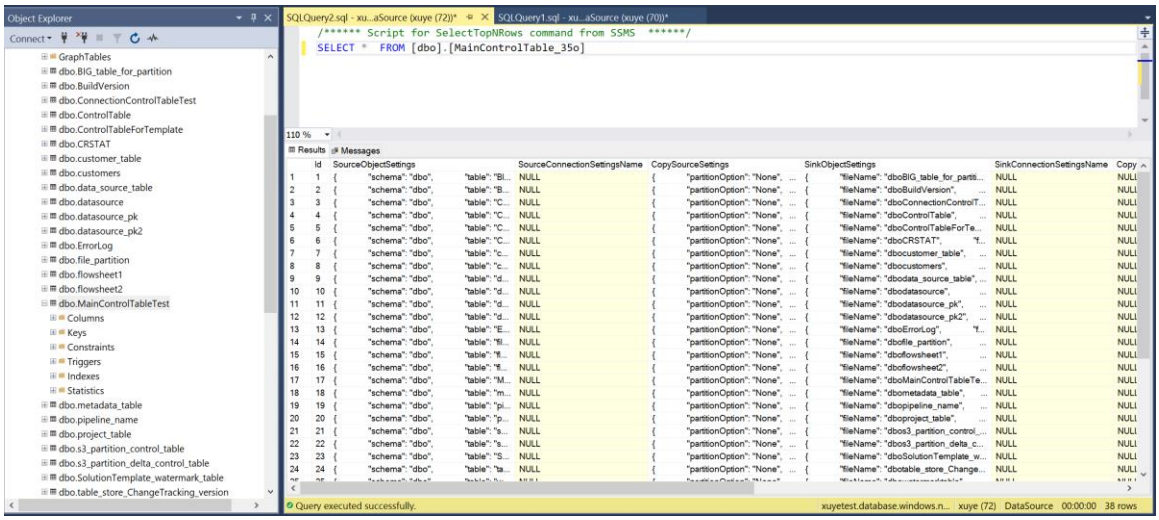
You will see 2 SQL scripts in total.

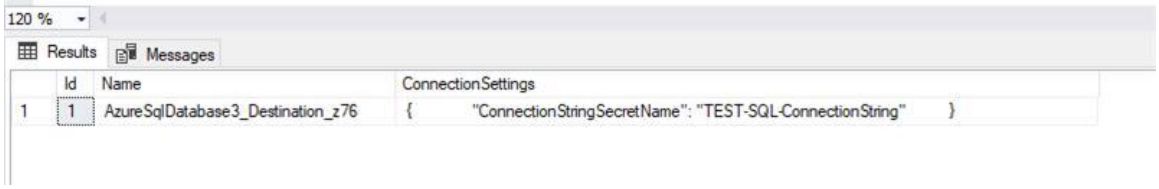
- The 1st SQL script is used to create a main control table and a connection control table. The metadata (table names etc.) will be stored in the main control table. The connection control table is used to store the connection string of your data store if you are using parametrized linked service in copy data tool.
- The 2rd SQL script is used to create a store procedure. It will update the latest watermark into control table every time from generated pipeline to your control table.

12. Open SSMS to connect to your control table server, and copy paste the scripts to create control tables as well as store procedure.

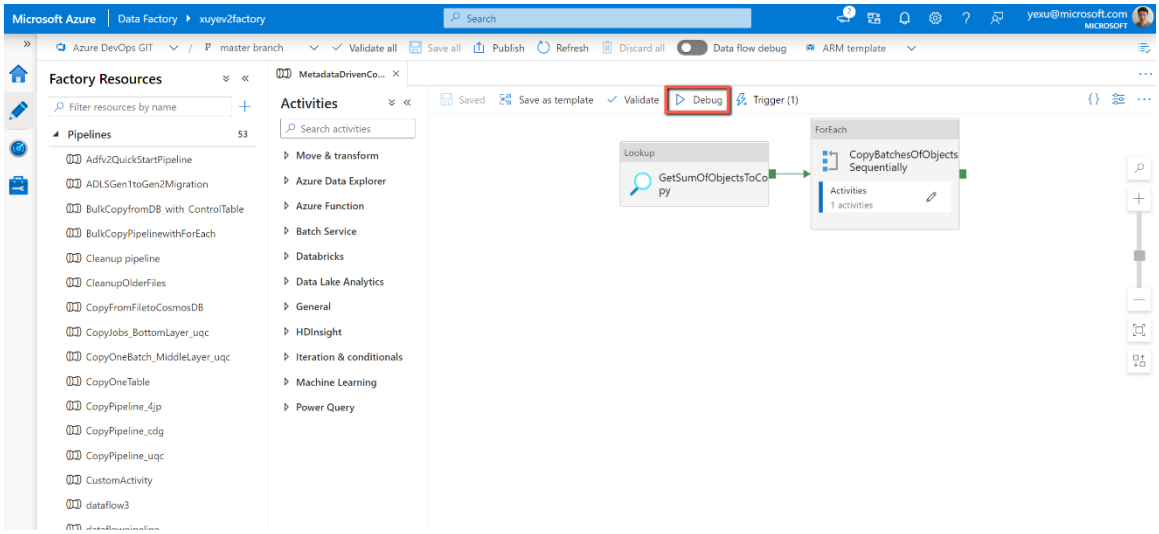


13. Query the control table to check if all the metadata is available or not.





14. Go back to ADF portal and **debug** the generated pipelines.



You will see you are required to input the following parameter:

Parameter:
Name: MaxNumberOfConcurrentTasks
Description: You can always change the max number of concurrent copy activity run before pipeline run. The default value will be the one you input in copy data tool.

Name: MainControlTableName

Description: You can always change the table name of main control table, so the pipeline will query the metadata from control table before run.

Name: ConnectionControlTableName
Description: You can always change the table name of connection control table (optional), so the pipeline will query the metadata related to data store connection before run.

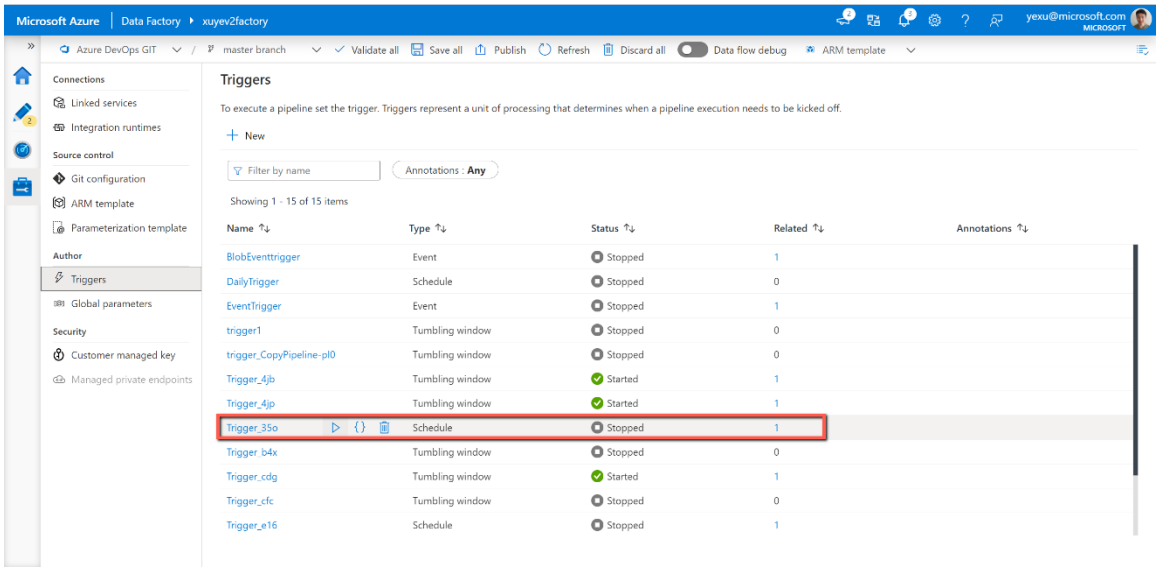
Name: MaxNumberOfObjectsReturnedFromLookupActivity
Description: In order to avoid reaching the limit of output lookup activity, there is a way to define the max number of object returned by lookup activity. In most case, the default value is not required to be changed.

Name: windowStart

Description: When you input dynamic value (e.g. yyyy/mm/dd) as folder path, the parameter is used to pass the current trigger time to pipeline in order to fill the dynamic folder path. When the pipeline is triggered by schedule trigger or tumbling windows trigger, user does not need to input the value of this parameter.

Sample value: 2021-01-25T01:49:28Z

15. Enable the trigger to operationalize the pipeline.

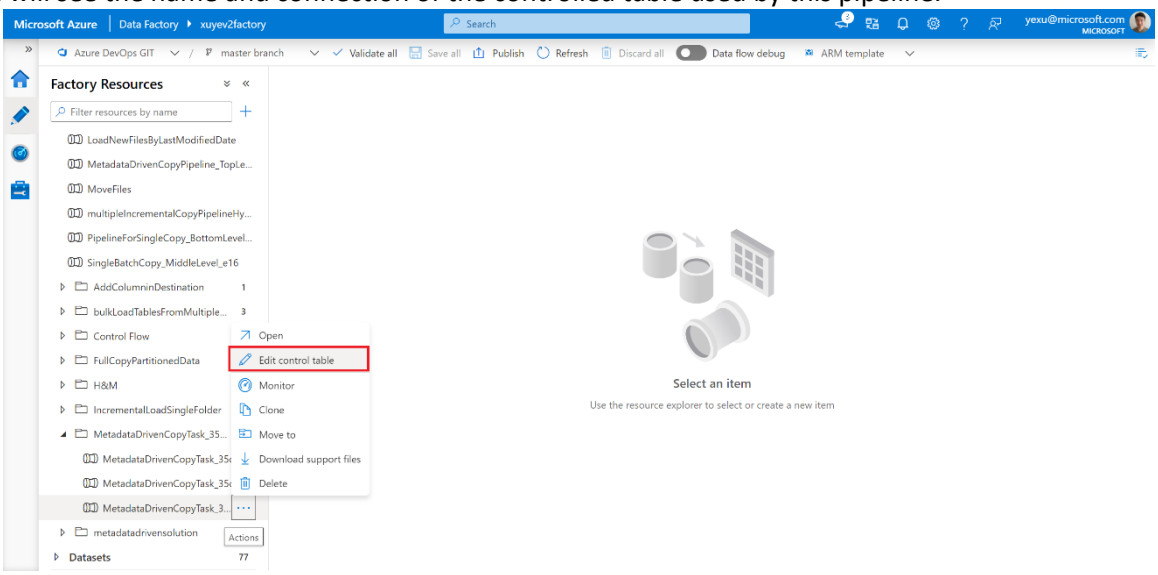


Edit control table:

You can directly update the control table to add or remove the object to be copied or change the copy behavior for each table. We also create a UI expereince in copy data tool to ease the jorney of editing the control table.

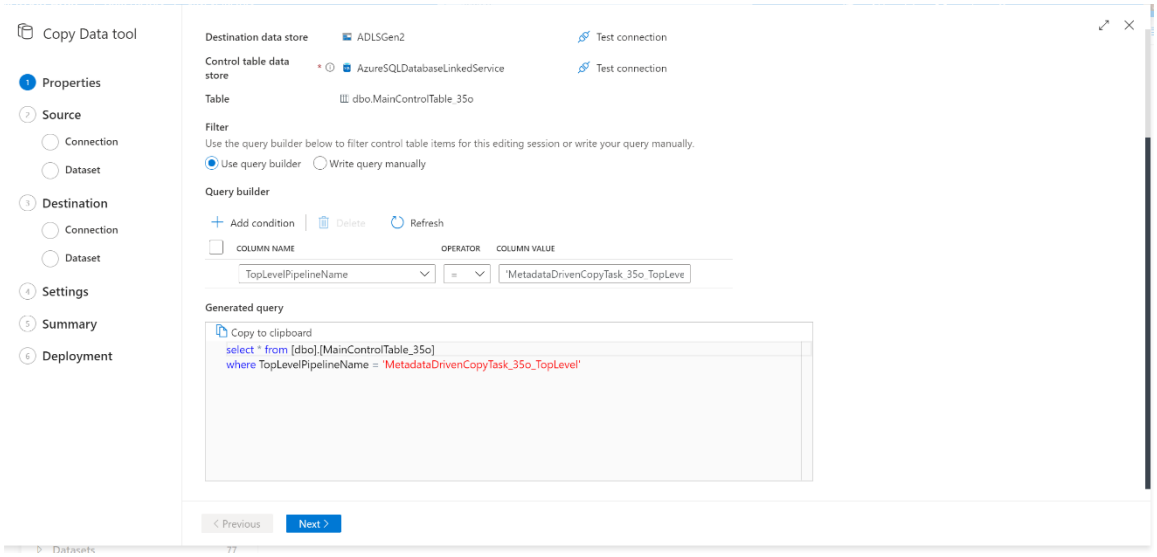
1. Right click the top level pipeline: **MetadataDrivenCopyTask_xxx_TopLevel,** and then select **edit control table.**

2. You will see the name and connection of the controlled table used by this pipeline.

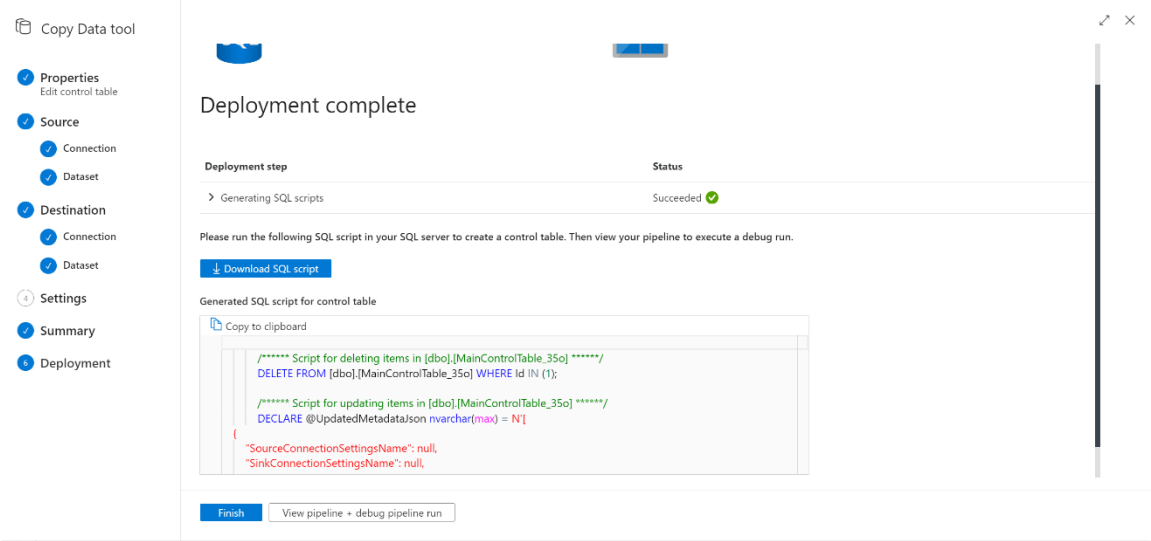


Compose the query to select objects from the control table to edit. By default, it will select all the tables copied by this pipeline via query:"

```
select * from [dbo].[MainControlTableName] where TopLevelPipelineName =
 'MetadataDrivenCopyTask_xxx_TopLevel' ".
```



3. You can add or remove more tables, change the column mapping etc. on copy data tool. After that, the copy data tool will come up with a new SQL script for you to update the control table.



Please note, the pipeline will **NOT** be redeployed. Only a SQL script will be created to help you to update the control table used by this pipeline.

Main control table description:

Each row in control table contains the metadata for one object (e.g. one table) to be copied

Column name	Description
`Id`	Unique ID of the object to be copied.
`SourceObjectSettings`	Metadata of source dataset. It can be schema name, table name etc. Example: Copy and transform data in Azure SQL Database - Azure Data Factory | Microsoft Docs
`SourceConnectionSettingsName`	It is optional. Name of the source connection setting in connection control table.
`CopySourceSettings`	Metadata of source property in copy activity. It can be query, partitions etc. Example: Copy and transform data in Azure SQL Database - Azure Data Factory | Microsoft Docs
`SinkObjectSettings`	Metadata of destination dataset. It can be file name, folder path, table name etc. If dynamic folder path specified, the variable value will not be written into control table. Example: Copy and transform data in Azure Data Lake Storage Gen2 - Azure Data Factory | Microsoft Docs
`SinkConnectionSettingsName`	It is optional. Name of the destination connection setting in connection control table.
`CopySinkSettings`	Metadata of sink property in copy activity. It can be preCopyScript, tableOption etc.

	Example: Copy and transform data in Azure SQL Database - Azure Data Factory | Microsoft Docs
`CopyActivitySettings`	Metadata of translator property in copy activity. It is used to define column mapping.
`TopLevelPipelineName`	Top Pipeline name which can copy this object.
`TriggerName`	Tigger name which can trigger the pipeline to copy this object.
`DataLoadingBehaviorSettings`	Full load vs. delta load.
`TaskId`	The order of objects to be copied following the TaskId in control table (ORDER BY [TaskId] DESC). If you have huge amounts of objects to be copied but only limited concurrent number of copied allowed, you can changed the TaskId for each object to decide which objects can be copied earlier. The default value for TaskID is 0 for all the objects in control table.

Connection Control Table:

Name	Name of the parameterized connection
ConnectionSettings	Value of the connection. It can be DB name, Server name etc.

Generated pipeline/activity description:

You will see 3 levels of pipelines are generated by copy data tool.

Top level Pipeline: MetadataDrivenCopyTask_xxx_TopLevel

Description: This pipeline will count the total number of objects (tables etc.) required to be copied in this run, come up with the number of sequential batches based on the max allowed concurrent copy task, and then execute another pipeline to copy different batches sequentially.

Parameters:

- **Name**: MaxNumberOfConcurrentTasks
- **Description:** You can decide the max number of concurrent copy activity run in order to control the load impact on your source store. The default value is 20. It means by default 20 copy activity runs in the generated pipeline will load data in parallel from your source database. You can always change the max number of concurrent copy activity run before pipeline run.

- **Name**: MainControlTableName
- **Description:** The table name for main control table. The pipeline will query the metadata from control table before run.

- **Name**: ConnectionControlTableName
- **Description**: The table name of connection control table (optional), so the pipeline will query the metadata related to data store connection before run.

- **Name**: MaxNumberOfObjectsReturnedFromLookupActivity
- **Description**: In order to avoid reaching the limit of output lookup activity, there is a way to define the max number of object returned by lookup activity. In most case, the default value is not required to be changed.

Activities:

- **Lookup activity:**
- **Name**: GetSumOfObjectsToCopy
- **Description:** Count the total number of objects (tables etc.) required to be copied in this run.

- **ForEach activity:**
- **Name**: CopyBatchesOfObjectsSequentially
- **Description:** Come up with the number of sequential batches based on the max allowed concurrent copy tasks, and then execute another pipeline to copy different batches sequentially.

- **Execute Pipeline activity:**
- **Name**: CopyObjectsInOneBtach
- **Description:** Execute another pipeline to copy one batch of objects. The objects belonging to this batch will be copied parallelly.

Middle level Pipeline: MetadataDrivenCopyTask_xxx_ MiddleLevel

Description: This pipeline will copy one batch of objects. The objects belonging to this batch will be copied parallelly.

Parameters:
- **Name**: MaxNumberOfObjectsReturnedFromLookupActivity
- **Description**: In order to avoid reaching the limit of output lookup activity, there is a way to define the max number of objects returned by lookup activity. In most case, the default value is not required to be changed.

- **Name**: TopLayerPipelineName
- **Description:** The name of top layer pipeline.

- **Name**: TriggerName
- **Description:** The name of trigger.

- **Name**: CurrentSequentialNumberOfBatch
- **Description:** The id of sequential batch.

- **Name**: SumOfObjectsToCopy
- **Description:** The total number of objects to copy.

- **Name**: SumOfObjectsToCopyForCurrentBatch
- **Description:** The number of objects to copy in current batch.

- **Name**: MainControlTableName
- **Description:** The name of main control table.

- **Name**: ConnectionControlTableName
- **Description:** The name of connection control table.

Activities
- **ForEach activity:**
- **Name**: DivideOneBatchIntoMultipleGroups
- **Description:** Divide objects from single batch into multiple sub parallel groups to avoid reaching the output limit of lookup activity.

- **Lookup activity:**
- **Name**: GetObjectsPerGroupToCopy
- **Description:** Get objects (tables etc.) from control table required to be copied in this group. The order of objects to be copied following the TaskId in control table (ORDER BY [TaskId] DESC).

- **Execute Pipeline activity:**
- **Name**: CopyObjectsInOneGroup
- **Description:** Execute another pipeline to copy objects from one group. The objects belonging to this group will be copied parallelly.

Bottom level pipeline: MetadataDrivenCopyTask_xxx_ BottomLevel

Description: This pipeline will copy objects from one group. The objects belonging to this group will be copied parallelly.

Parameters:
- **Name**: ObjectsPerGroupToCopy
- **Description**: The number of objects to copy in current group.

- **Name**: ConnectionControlTableName
- **Description**: The name of connection control table.

- **Name:** windowStart
- **Description:** It used to pass the current trigger time to pipeline in order to fill the dynamic folder path if configured by user.

Activities:
- **ForEach activity:**

- **Name**: ListObjectsFromOneGroup
- **Description:** List objects from one group and iterate each of them to downstream activities.

- **Switch activity:**
- **Name**: RouteJobsBasedOnLoadingBehavior
- **Description:** Check the loading behavior for each object if it requires full load or incremental load. If it is default or FullLoad case, do full load. If it is DeltaLoad case, do incremental load via watermark column to identify changes.

- **Copy activity** (in default or full load case):
- **Name**: FullLoadOneObject
- **Description:** Take a full snapshot on this object and copy it to the destination.

- **Lookup activity** (in delta load case):
- **Name**: GetMaxWatermarkValue
- **Description:** Query the source object to get the max value from watermark column.

- **Copy activity** (in delta load case):
- **Name**: DeltaLoadOneObject
- **Description:** Copy the changed data only from last time via comparing the value in watermark column to identify changes.

- **StoreProcedure activity** (in delta load case):
- **Name**: UpdateWatermarkColumnValue
- **Description:** Write back the new watermark value to control table to be used next time.

Know limitation:

- It only supports database as source store now. You can further manually build parameterized pipelines for file-based store as well.

- IR name, database type, file format type cannot be parameterized in ADF. For example, if you want to ingest data from both Oracle Server and SQL Server, you will need 2 different parameterized pipelines. But the control table can be shared.