TIP SHEET

Azure Data Factory Parameterization

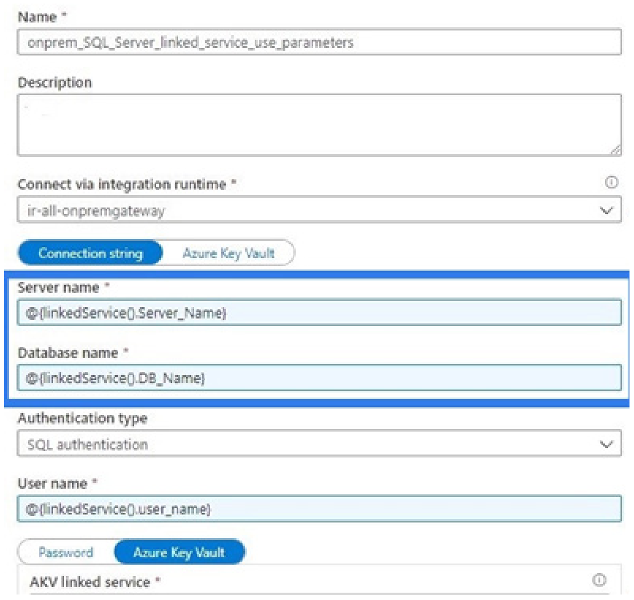
By implementing parameters in Azure Data Factory (ADF) development, you can save time by minimizing the amount of hard coding needed for additional connections or ETL processes. Parameters add flexibility to your solution and allow your ETL solutions to scale naturally. For example, instead of creating a dataset object for every SQL table on a database, developers can create just one reusable dataset for the SQL connection and reduce a data swamp of ADF objects. Additionally, consider if a new table to the solution follows the same ETL pattern; with parameters established, the pipeline can be dynamic and flexible to process the additional table with minimal changes to the existing solution.

To attain the full benefits of parameters in ADF, check out the following tips that can be utilized to maximize your full ADF potential and achieve a scalable, reliable ETL platform. These tips will break down per resource what can and should be parameterized to maximize ADF's value.

Linked Services

Linked Services define the connection information from the Azure service to external resources. ADF allows users to pass dynamic values at runtime. For example, if you have multiple databases on the same logical SQL server, you can create a parameter for the database name in the linked service definition. Adding this parameter means you only need to manage one connection for all the databases, instead of a single linked service for every database you wish to connect to.

For the most secure solution, it is recommended to use the Linked Service parameter option with the Azure Key Vault connection. In Key Vault, you can store the connection string information for your connection, including all sensitive credential information that should be secure. Then, when creating your linked service, you can set up a parameter for the 'Secret Name' to reference the secret in Key Vault and it will retrieve your connection string securely, with reusability across other resource connections.


Name *
onprem_SQL_Server_linked_service_use_parameters

Description

Connect via integration runtime *
ir-all-onpremgateway

Server name *
@{linkedService().Server_Name}

Database name *
@{linkedService().DB_Name}

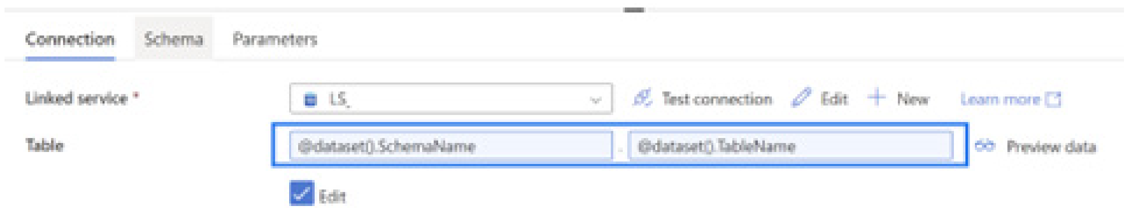
Authentication type
SQL authentication

User name *
@{linkedService().user_name}

AKV linked service *

Datasets

Once you create a Linked Service which authenticates access to your data, you can create a dataset. The dataset serves to connect to a specific data object via the credential information provided by the Linked Service. Inside the dataset, you can add parameters for connection information such as, ‘SchemaName’, or ‘TableName’ for a SQL connection, and at runtime you can enter the values for the data you want to use. Or for Data Lake storage, you can create parameters like ‘ContainerName’, ‘DirectoryPath’, or ‘FileName’ and can access the files you need by entering the values. This saves you time and resources. If you have 100 tables to use, you don’t need to create 99 more datasets, you can use parameters to make one dataset apply for each table.



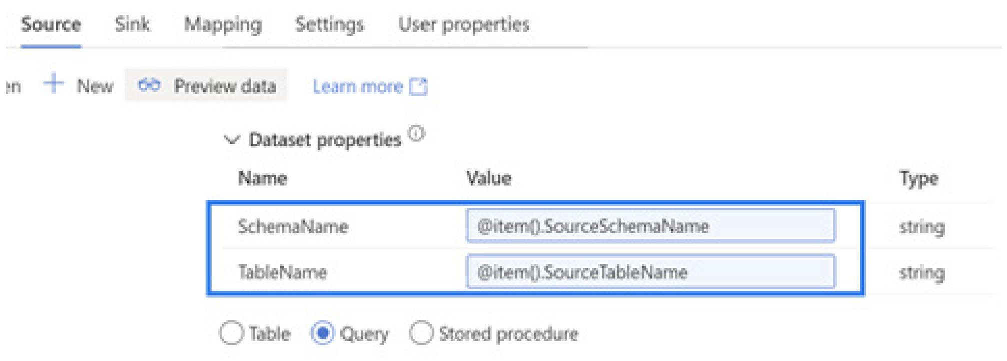
Linked service * LS_

Table @dataset().SchemaName . @dataset().TableName

☑ Edit

Pipelines

Pipelines are a logical grouping of activities that perform a task and are used to orchestrate ETL jobs. You can use parameters in a pipeline to make it reusable for multiple tables. For example, if you are using a copy data activity for multiple tables, you can create parameters for the Source and Sink values in your pipeline. Then at runtime, you can choose the files you are copying from and the table you are populating. This allows the pipeline to be flexible, and scale as additional files or tables are added to the solution.



Dataset properties

Name	Value	Type
SchemaName	@item().SourceSchemaName	string
TableName	@item().SourceTableName	string

Table [✓ Query] Stored procedure