Hands-On Lab: Passing Parameter and Variable Values from Pipeline to Mapping Data Flow in Azure Data Factory

Author: Sandeep Kumar Sharma

Learning Objective

This lab teaches you how to pass values from a Pipeline (using Parameters and Variables) into a Mapping Data Flow in Azure Data Factory. You will learn how to define parameters in a data flow, set values from the pipeline, and observe how the values are used in transformations.

Learning Outcome

After completing this lab, you will be able to: - Create Pipeline Parameters and Variables - Create Data Flow Parameters - Pass values from Pipeline to Data Flow - Use passed values inside Mapping Data Flow expressions

---

1. Introduction

In Azure Data Factory, a Pipeline can send values to a Mapping Data Flow. These values can be dynamic, taken from parameters or variables inside the pipeline. This is useful when you want to make your data flow reusable, dynamic, and controlled at runtime.

In this exercise, we will: - Create a pipeline with a parameter and a variable - Create a mapping data flow with a parameter - Pass values from pipeline → data flow - Use that value inside a Derived Column transformation

---

2. Prerequisites

Make sure you have: - An Azure Data Factory instance - A simple CSV dataset in storage (with a column like `name`) - Basic understanding of pipelines and data flows

---

3. Step-by-Step Hands-On Lab

Step 1: Open Azure Data Factory Studio

1. Go to Azure Portal.
2. Open Azure Data Factory.
3. Click **Launch Studio**.

Step 2: Create a Mapping Data Flow

1. Go to **Author**.
2. Click **+** → **Data Flow**.
3. Select **Mapping Data Flow**.
4. Name it: `ParameterDataFlow`.

Add a Data Flow Parameter

1. In the Data Flow, select **Parameters** (top menu).
2. Click **New**.
3. Create a parameter:
4. Name: `prefixValue`
5. Type: `String`

This parameter will receive value from the pipeline.

Add a Source

1. Click **Add Source**.
2. Select your dataset.
3. Confirm preview after enabling debug (optional).

Add a Derived Column

1. Click **+** after the Source.
2. Choose **Derived Column**.
3. Add a new column:
4. Column Name: `final_name`
5. Expression:

   ```
   prefixValue ~ name
   ```

   This will add the prefix value to the existing name column.

Save the Data Flow.

Step 3: Create a New Pipeline

1. Go to **Author**.
2. Click **+** → **Pipeline**.
3. Name the pipeline: `ParameterPipeline`.

Add a Pipeline Parameter

1. Click **Parameters** (top).
2. Click **New**.
3. Add:
4. Name: `pipelinePrefix`
5. Type: `String`
6. Default Value: `EMP_`

Add a Pipeline Variable

1. Click **Variables**.
2. Click **New**.
3. Add:
4. Name: `suffixVar`
5. Type: `String`
6. Default Value: `_2025`

Now we have: - A pipeline parameter: `pipelinePrefix` - A pipeline variable: `suffixVar`

Step 4: Add a Data Flow Activity

1. In the pipeline, click **Activities → Move & Transform**.
2. Drag **Data Flow** onto the canvas.
3. Select the **ParameterDataFlow** you created earlier.

Pass Pipeline Parameter to Data Flow Parameter

1. In the Data Flow activity → go to **Parameters**.
2. You will see the data flow parameter `prefixValue`.
3. Set its value:

```
@concat(pipeline().parameters.pipelinePrefix, variables('suffixVar'))
```

This means: - First part comes from pipeline parameter - Second part comes from pipeline variable

Example result: `EMP__2025`

Step 5: Validate and Debug

1. Click **Debug**.
2. Enter a value for pipeline parameter `pipelinePrefix` if required.
3. Run the pipeline.

The Mapping Data Flow will receive the complete string like:

```
EMP__2025
```

And apply it inside `final_name` column.

---

4. Final Description

This lab demonstrated how a pipeline can send values to a Mapping Data Flow by using parameters and variables. You created parameters in both pipeline and data flow, passed combined dynamic values, and used them inside a Derived Column expression.

This is a powerful technique in Azure Data Factory for creating dynamic and reusable workflows.