Hands-On Lab: Convert CSV File to JSON File Using Mapping Data Flow in Azure Data Factory

Author: Sandeep Kumar Sharma

Learning Objective

This lab teaches you how to convert a CSV file into a JSON file using the **Mapping Data Flow** activity in Azure Data Factory. The lab is designed in a beginner-friendly, easy step-by-step format.

Learning Outcome

After completing this lab, you will be able to: - Read a CSV file using a dataset - Configure a Mapping Data Flow to transform CSV → JSON format - Add a Sink to write JSON output to Azure Storage - Run the Data Flow from a pipeline

---

1. Introduction

Many real-time data integration scenarios require converting data from CSV format to JSON format. Azure Data Factory Mapping Data Flow makes this possible without writing code — using transformations and sink configuration.

In this lab, you will: - Use a CSV file as input - Transform it using Mapping Data Flow - Produce output as a JSON file in Azure Data Lake or Blob Storage

---

2. Dataset Used in This Lab

You will upload the following CSV file to ADLS/Blob Storage.

CSV File: `EmployeeData.csv`

EmpID	EmpName	Department
1	John	HR
2	Priya	Finance
3	Michael	IT

This CSV will be converted to a JSON file.

3. Step-by-Step Hands-On Lab

Step 1: Upload CSV File to Azure Storage

1. Open your Storage Account.
2. Create a new folder: `input/`.
3. Upload the file `EmployeeData.csv`.

Step 2: Create a CSV Dataset in ADF

1. Open **Azure Data Factory Studio**.
2. Go to **Author → Datasets**.
3. Click **+ New Dataset**.
4. Select **Azure Data Lake Storage Gen2** or **Blob Storage**.
5. Choose **CSV**.
6. Name the dataset **EmployeeCSVInput**.
7. In file path:
8. Folder: `input/`
9. File: `EmployeeData.csv`
10. Enable **First row as header**.
11. Save the dataset.

Step 3: Create a Mapping Data Flow

1. Click **+** → **Data Flow**.
2. Select **Mapping Data Flow**.
3. Name it: **CSVToJSONDataFlow**.

Step 4: Add Source Transformation

1. Click **Add Source**.
2. Select dataset: **EmployeeCSVInput**.
3. Enable:
4. **Allow schema drift**
5. **Infer drifted column types**

You will see the CSV columns when using Data Preview (after enabling debug).

Step 5: Add Sink Transformation for JSON Output

Now configure the output to write JSON.

1. Click **+** after Source.
2. Select **Sink**.
3. Choose **Azure Data Lake Storage Gen2** or **Blob Storage**.
4. Create a new dataset:
5. Type: **JSON**
6. Name: **EmployeeJSONOutput**
7. Folder: `output/`
8. File name: leave blank (ADF will generate automatically)
9. In Sink Settings:
10. Set **File Format** = JSON
11. Set **Single File Output** = Yes (if required)
12. Enable **Overwrite** if you want to replace existing file

ADF will now generate a JSON file.

Step 6: Debug and Preview Output

1. Turn on **Data Flow Debug**.
2. Go to **Data Preview**.
3. You will see the source CSV rows.

The final output JSON will look like:

```
{
  "EmpID": 1,
  "EmpName": "John",
  "Department": "HR"
}
{
  "EmpID": 2,
  "EmpName": "Priya",
  "Department": "Finance"
}
{
  "EmpID": 3,
  "EmpName": "Michael",
  "Department": "IT"
}
```

Step 7: Execute the Data Flow through a Pipeline

1. Create a new Pipeline.
2. Drag **Data Flow** activity to pipeline canvas.
3. Select **CSVToJSONDataFlow**.
4. Validate and click **Debug**.

Once executed, go to your storage account: - Open the `output/` folder - You will see a JSON file generated (example: `part-0000.json`)

---

4. Final Description

In this lab, you learned how to use Azure Data Factory's Mapping Data Flow to convert a CSV file into JSON format. You created a source for CSV, configured a JSON sink, and executed the job using a pipeline. This method is commonly used for file format conversions, API ingestion, and NoSQL database loading.

This completes the hands-on exercise.