**Use Case: Azure File Processing with Control File and File Count Validation**

**Business Requirement**

A company ingests daily data files from an external system into Azure Blob Storage. Before processing these files, the system needs to ensure that:

The company receives daily data files uploaded to **Azure Blob Storage**. . Before processing these files, the system needs to ensure that:

1. Validate that the **number of records** in each file matches the count specified in a **control CSV file**.
2. Ensure that the file names match those listed in the control file.
3. If the record counts and file names are valid, the files are processed and loaded into the target system (e.g., Azure SQL Database or Data Lake).
4. If any validation fails, the process stops, logs the error, and sends an email notification.

**Scenario Overview**

**The external system uploads a control file and several data files to an Azure Blob Storage container. The control file contains metadata about the expected files (e.g., file names, count). The Azure pipeline will validate the files against the control file before loading them into a target destination. This ensures that no files are missed or processed incorrectly, which is critical for ensuring data integrity.**

**CSV Control File Example**

The control file (control\_file\_2024\_09\_15.csv) will look like this:

|  |  |
| --- | --- |
| file\_name | record\_count |
| sales\_data\_2024\_09\_15.csv | 1000 |
| inventory\_data\_2024\_09\_15.csv | 500 |
| customer\_data\_2024\_09\_15.csv | 1500 |

In this file:

* file\_name: Specifies the name of each file.
* record\_count: Specifies the expected number of records in each file.

**Solution Hints:**

**1. Upload Data Files and Control File to Azure Blob Storage**

The external system uploads:

* A **control file** (e.g., control\_file\_2024\_09\_15.csv) to **Azure Blob Storage**.
* **Actual data files**, which will be processed only if validation passes.

**Control File Format (CSV)**

The control file will contain:

* The expected **file names**.
* The expected **record counts** for each file.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

In this example, the control file specifies:

* **sales\_data\_2024\_09\_15.csv**: expected to have 1000 records.
* **inventory\_data\_2024\_09\_15.csv**: expected to have 500 records.
* **customer\_data\_2024\_09\_15.csv**: expected to have 1500 records.

**2. Azure Data Factory Pipeline Setup**

The **Azure Data Factory (ADF)** pipeline will automate the following tasks:

* **Get Control File**: Use the **Lookup Activity** to read the control CSV file from Blob Storage.
* **Get Actual Files**: Use the **Get Metadata Activity** to fetch the actual files from Blob Storage.
* **Check Record Count**: Use **Data Flow** or **Custom Script** (such as PySpark or Azure Function) to validate the record counts in the files.
* **Compare with Control File**: Validate that the actual record counts in each file match those specified in the control file.

**3. File Name Validation**

* Compare the file names listed in the control file with the actual file names in Blob Storage.
* If there is a mismatch, the pipeline should stop processing and log the issue.

**4. Record Count Validation**

* For each file listed in the control file, retrieve the actual **record count** using either:
  + **Azure Data Flow**: It can count the number of rows in each file.
  + **Custom Activity**: A PySpark or Azure Function script to read the file and count the rows.
* Compare the actual record count with the record\_count specified in the control file.

**5. Process Files (If Validation Passes)**

* If both the file name and record count validations pass, the files can be processed (e.g., copied to an Azure SQL Database or transformed using Azure Databricks).

**6. Error Handling and Logging**

* If there’s a mismatch (in either file name or record count), the pipeline will stop processing, log the error to a table in **Azure SQL Database**, and send an email notification using the **Send Email Activity** in Azure Data Factory.

**Technical Flow for Record Count Validation**

1. **Extract Control File Information**:
   * Use **Lookup Activity** to read the control file (control\_file\_2024\_09\_15.csv), which contains the file\_name and record\_count.
2. **Get Actual Files**:
   * Use **Get Metadata Activity** to list the actual files in the Blob container.
3. **File Name Validation**:
   * Compare the file names listed in the control file with the actual files in Blob Storage. If they don’t match, stop the pipeline and log the error.
4. **Record Count Validation**:
   * Use **Data Flow** or a **Custom Activity** (PySpark or Azure Function) to count the records in each file.
   * Compare the actual record count with the count specified in the control file. If they don’t match, stop the pipeline and log the issue.
5. **Process Files**:
   * If validation passes, process the files (e.g., copy them into an Azure SQL Database or Data Lake, or transform them using Databricks).

**Error Handling and Logging**

In case of a failure:

1. The pipeline will log the error details (file name mismatch, incorrect record count, etc.) in an **Azure SQL Database** table.
2. The **Send Email Activity** will trigger an email notification, alerting the team about the validation failure.

**Benefits**

1. **Data Integrity**: Validates that the actual number of records in the file matches the expected number, ensuring that no partial or incorrect data is processed.
2. **Automation**: Automates file and record count validation, reducing the risk of manual errors.
3. **Error Tracking**: Logs errors for review and sends email alerts to ensure that issues are addressed promptly.
4. **Scalability**: Can handle large datasets and files, making it suitable for various ETL pipelines.
5. **Customization**: Can be extended to perform additional validation (e.g., schema validation, checksum validation).

This approach is ideal for use cases where the number of records in the data files must match a control file, ensuring accurate data processing in the ETL pipeline.