SSIS Foreach Loop Container (File Enumerator): Load Multiple CSVs into One SQL Table and Log Row Counts

Role: SSIS Integration Specialist Trainer — clear, step-by-step notes for Visual Studio 2022 / SSDT.

# 1) What is the Foreach Loop Container?

The Foreach Loop Container repeats the tasks inside it for each item in a collection. With the Foreach File Enumerator, that collection is the list of files in a folder.

Use it when you need to process many files with the same logic, e.g., load every daily CSV into the same table, or archive a set of logs.

# 2) Real-world Scenario

You receive 10+ CSVs in C:\files every day (customer\_1.csv, customer\_2.csv, …). You must load all files into a single SQL Server table (dbo.PersonData) and, for each file, log the file path and number of rows into dbo.FileInfo.

# 3) Destination Tables (create once in SQL Server)

Example schemas (adjust column names/types to match your CSV):

CREATE TABLE dbo.PersonData (  
 ID INT,  
 FirstName NVARCHAR(50),  
 LastName NVARCHAR(50),  
 Gender NVARCHAR(10),  
 CompanyName NVARCHAR(100)  
);  
  
CREATE TABLE dbo.FileInfo (  
 id INT IDENTITY(1,1) PRIMARY KEY,  
 filepath NVARCHAR(260),  
 recordcount INT,  
 dated DATETIME DEFAULT(GETDATE())  
);

# 4) Package Variables (SSIS → Variables)

Create the following User scope variables:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Initial Value | Purpose |
| FilePath | String |  | Holds the current file full path during the loop. |
| Cnt | Int32 | 0 | Holds the row count for the current file (set by Row Count transform). |
| SourceFolder | String | C:\files | Folder containing the CSVs. |
| FilePattern | String | \*.csv | Wildcard filter, e.g., \*.csv or \*\_2017.txt. |

# 5) Control Flow Layout (VS 2022 / SSDT)

From SSIS Toolbox, drag these onto the Control Flow surface:

• Foreach Loop Container (we will configure it next)  
 – Inside the container place:  
 1) Data Flow Task → name it “Load Data”  
 2) Execute SQL Task → name it “Log File Info”  
Connect the Data Flow Task to the Execute SQL Task with a green precedence constraint.

# 6) Data Flow: Load One File

Open Load Data (double-click) and configure:

1) Flat File Source  
 • Click New… to create a Flat File Connection Manager. Pick any sample CSV (e.g., customer\_1.csv) to infer columns.  
 • Set Text qualifier to " if your data is quoted.  
 • Ensure column delimiters and header row are correct.

2) Row Count transformation  
 • Drag Row Count between the Flat File Source and destination.  
 • Double-click → VariableName = User::Cnt (captures rows processed for this file).

3) OLE DB Destination  
 • Create/choose a connection manager to your database.  
 • Data access mode: Table or view – fast load.  
 • Name of the table: dbo.PersonData (or your table).  
 • Click Mappings and ensure all columns align.

4) Make the Flat File Connection dynamic  
 • Select the Flat File Connection Manager (bottom pane).  
 • In Properties → Expressions… → set ConnectionString to:

@[User::FilePath]

• Set DelayValidation = True on the Data Flow Task and on the Flat File Source to avoid design-time file checks.

# 7) Execute SQL Task: Log File Path + Record Count

Place this task after the Data Flow inside the Foreach container. Two safe ways:

Option A (Recommended — Parameterized)

INSERT INTO dbo.FileInfo (filepath, recordcount, dated)  
VALUES (?, ?, GETDATE());  
  
Parameter Mapping:  
 0 → VariableName=User::FilePath, Direction=Input, DataType=NVARCHAR, Size=260  
 1 → VariableName=User::Cnt, Direction=Input, DataType=INT32

Option B (Expression-based quick method) — set Expressions → SQLStatementSource to:

"INSERT INTO dbo.FileInfo (filepath, recordcount, dated) VALUES ('"  
+ REPLACE(@[User::FilePath], "'", "''")  
+ "', " + (DT\_STR,12,1252)@[User::Cnt] + ", GETDATE())"

# 8) Configure the Foreach Loop Container

Right-click the Foreach container → Edit

Collection tab:

• Enumerator: Foreach File Enumerator  
• Folder: C:\files (or set via Expressions using @[User::SourceFolder])  
• Files: \*.csv (or @[User::FilePattern] via Expressions)  
• Retrieve file name: Fully qualified  
• (Optional) Uncheck Traverse subfolders unless you need recursion

Variable Mappings tab:  
• Index 0 → User::FilePath

# 9) Run and Validate

-- After execution, validate with:  
SELECT COUNT(\*) AS PersonDataRows FROM dbo.PersonData;  
SELECT \* FROM dbo.FileInfo ORDER BY id DESC;

# 10) Tips, Patterns, and When to Use

Use Foreach File when: many files share the same schema and must be processed identically; you need daily automation; or you want per-file audit rows.

Best practices:  
• DelayValidation = True where paths change at runtime.  
• Prefer parameterized SQL for logging/updates.  
• Place Row Count right before the destination.  
• Add error redirection for bad rows.  
• Archive/move files after success to avoid reprocessing.

Common pitfalls:  
• Wrong Retrieve file name mode (use Fully qualified).  
• ConnectionString expression not set to User::FilePath.  
• Parameter data type mismatches in Execute SQL Task.