**SSIS Row Count Transformation – End-to-End Guide + Fix Checklist (Foreach CSV → SQL + Audit Logs)**

# 1) What the Row Count Transformation Does

Row Count counts rows flowing through a path and stores the result in an Int32 SSIS variable. Typical uses: per‑file auditing, totals, emails, conditional flows.

# 2) Scenario

Load multiple CSV files (TestData\_1..3.csv) into dbo.TestData and write FilePath, RecordCount, Dated into dbo.Logs. Header row is ignored.

# 3) SQL – Destination and Audit Tables

```sql  
CREATE TABLE dbo.TestData(  
 ID varchar(50) NULL,  
 first\_name varchar(50) NULL,  
 last\_name varchar(50) NULL,  
 gender varchar(50) NULL,  
 company\_name varchar(50) NULL  
);  
  
CREATE TABLE dbo.Logs(  
 ID int IDENTITY(1,1) PRIMARY KEY,  
 FilePath varchar(300) NOT NULL,  
 RecordCount int NOT NULL,  
 Dated datetime NOT NULL DEFAULT(getdate())  
);  
```

# 4) SSIS Variables

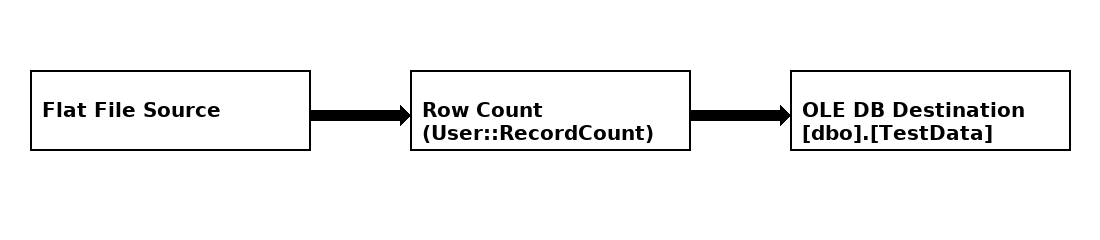
* User::FilePath (String): full path of current file
* User::RecordCount (Int32): set by Row Count transform

# 5) Control Flow – Foreach File

1. Add \*\*Foreach Loop Container\*\*.
2. Collection: \*\*Foreach File Enumerator\*\*, Folder = your CSV folder, Files = `\*.csv`, Retrieve = \*\*Fully qualified\*\*.
3. Variable Mappings: Index 0 → \*\*User::FilePath\*\*.
4. Inside: a \*\*Data Flow Task\*\* then an \*\*Execute SQL Task\*\* (Success precedence).

# 6) Data Flow – Source → Row Count → Destination

1. Add \*\*Flat File Source\*\*. Create a Flat File Connection Manager using one CSV for metadata; check \*\*Column names in the first data row\*\*.
2. Bind it to the variable: Connection Manager → \*\*Properties → Expressions → ConnectionString = @[User::FilePath]\*\*.
3. Set \*\*Data Flow Task → DelayValidation = True\*\* and \*\*Flat File Source → ValidateExternalMetadata = False\*\*.
4. Add \*\*Row Count\*\* and set \*\*VariableName = User::RecordCount\*\*.
5. Add \*\*OLE DB Destination\*\* → map to dbo.TestData.



# 7) Execute SQL Task – Parameterized INSERT

1. ConnectionType = OLE DB
2. SQLStatement: `INSERT INTO dbo.Logs(FilePath, RecordCount, Dated) VALUES (?, ?, GETDATE());`
3. Parameter 0: \*\*User::FilePath\*\*, VARCHAR(300)
4. Parameter 1: \*\*User::RecordCount\*\*, LONG (Int32)

# 8) Verify & Email (Optional)

Add a Script Task (ReadOnlyVariables: User::FilePath, User::RecordCount) between Data Flow and Execute SQL to log values. Optionally, add a Send Mail Task to email the count per file.

```csharp  
bool fire=false;  
Dts.Events.FireInformation(0,"Debug",  
 $"File={Dts.Variables[\"User::FilePath\"].Value}; Rows={Dts.Variables[\"User::RecordCount\"].Value}","",0,ref fire);  
```

# 9) Fix Checklist for Your Package (27 RowCount TransformationDemo.dtsx)

* Flat File \*\*Connection Manager\*\* has \*\*Expressions → ConnectionString = @[User::FilePath]\*\*.
* Data Flow Task \*\*DelayValidation = True\*\*; Flat File Source \*\*ValidateExternalMetadata = False\*\*.
* Row Count sits between Source and Destination and writes to \*\*User::RecordCount (Int32, package scope)\*\*.
* Execute SQL Task uses \*\*parameters\*\* (not an expression string) and is connected with \*\*Success\*\* from the Data Flow.
* Foreach enumerator returns \*\*Fully qualified\*\* file names and maps \*\*Index 0 → User::FilePath\*\*.
* There is only \*\*one\*\* Flat File Connection Manager referenced by the source (delete duplicates).

# 10) Common Pitfalls

* Expression set on the \*\*source\*\* instead of the \*\*connection manager\*\* (source keeps reading the same file).
* Wrong variable bound (e.g., `User::RecordCount`) leading to errors like “Cannot open datafile '0'”.
* Reading `User::RecordCount` too early; fix with parameterized SQL and Success precedence.

# 1) Flat File Connection Manager → Expression: ConnectionString = @[User::FilePath]

**What it is:** You’re binding the file path of the connection **manager** to a variable.  
**Why:** In a Foreach loop you want to read a different file each iteration. Setting this expression means the connection points to whatever User::FilePath is at runtime (e.g., C:\Data\TestData\_1.csv, then ...\_2.csv, etc.).  
**Without it:** The source keeps reading the same file you picked at design time.  
**Tip:** Put the expression on the **connection manager**, not on the Flat File **Source**. That’s the classic reason for “it logs new names but always loads the first file.”

# 2) Flat File Source → ValidateExternalMetadata = False

**What it is:** Tells the source **not** to verify the external file/columns at design-time or pre-validate.  
**Why:** With a dynamic ConnectionString, the specific file may not exist yet (or may change) when SSIS tries to validate. Setting this to **False** avoids early failures like “cannot open datafile…”.  
**Trade-off:** You give up a bit of early safety. If the file is truly missing or malformed, you’ll discover it **at runtime** (which is fine in a Foreach scenario).  
**When to turn back ON:** Fixed paths, stable schemas—validation helps catch mistakes earlier.

# 3) Data Flow Task → DelayValidation = True

**What it is:** Defers the task’s validation until it actually runs.  
**Why:** Similar reason—when connections/queries are built from variables or expressions, early validation can fail because the objects aren’t ready yet. Delaying validation lets your Foreach set User::FilePath first, then the Data Flow validates against the **current** file.  
**Without it:** The package can fail before the loop assigns the right path, or it sticks to the first evaluated file.

# 4) Row Count → writes to User::RecordCount (Int32)

**What it is:** A transform that **counts rows flowing through it** and stores that count into the variable you choose.  
**When it updates:** During the Data Flow execution; the variable holds the **final** count once the flow finishes.  
**Why:** Perfect for per-file auditing—after the Data Flow completes, you can read User::RecordCount (e.g., in an Execute SQL Task) and log the file name + row count.  
**Gotcha:** If you build the INSERT text via **expression** too early, you might read 0. Using **parameterized SQL** (or placing the log task after the Data Flow with a Success constraint) ensures you capture the updated value.

 **Expression on Connection Manager** → “Which file should I read **this** iteration?”

 **ValidateExternalMetadata = False** → “Don’t check the file until I actually run.”

 **DelayValidation = True** → “Don’t validate this Data Flow until the loop sets the right values.”

 **Row Count → variable** → “Count the rows that actually flowed and save that number for logging.”