SSIS

Wednesday, February 12, 2025 11:39 AM

Table of Contents

SSIS (SQL Server Integration Services) Tasks in Control Flow

In **SSIS**, a **task** is a unit of work that performs a specific action in the **Control Flow** of an SSIS package. The **Control Flow** defines the execution workflow by using tasks and precedence constraints.

1. Types of SSIS Tasks & When to Use Them

◈ Data Flow Task

✅ **Purpose:** Moves and transforms data between sources and destinations.
✅ **When to Use:**

- Extracting data from multiple sources (SQL, Excel, Flat Files, APIs).
- Performing transformations (e.g., filtering, aggregating, merging).
- Loading data into a database or data warehouse.

📌 **Example:** ETL process to load customer sales data into a data warehouse.

🔹 Execute SQL Task

✅ **Purpose:** Runs SQL queries or stored procedures.

✅ **When to Use:**

- Creating/dropping tables before or after data processing.
- Running stored procedures for data validation or transformations.
- Inserting, updating, or deleting records.

📌 **Example:** Deleting old log records from a table before inserting new data.

🔹 File System Task

✅ **Purpose:** Manages files and folders (copy, move, delete, create).

✅ **When to Use:**

- Moving processed files to an archive folder.
- Deleting temporary files after execution.
- Creating directories dynamically.

📌 **Example:** Move successfully processed CSV files from an **input** folder to an **archive** folder.

🔹 Script Task

✅ **Purpose:** Executes custom **C#** or **VB.NET** code for complex logic.

✅ **When to Use:**

- Performing advanced calculations not possible with built-in SSIS tasks.
- Sending API requests or handling dynamic file names.
- Custom error handling.

📌 **Example:** Call an external API to fetch exchange rates before loading financial data.

🔹 Send Mail Task

✅ **Purpose:** Sends email notifications.

✅ **When to Use:**

- Sending alerts for successful or failed package execution.
- Notifying users when a data load is complete.

📌 **Example:** Send an email when an ETL process fails with the error details.

🔹 Execute Process Task

✅ **Purpose:** Runs an external application or batch file.

✅ **When to Use:**

- Running a **PowerShell script** or **command-line utility**.
- Executing **third-party applications** as part of the workflow.

📌 **Example:** Run a .bat file to compress files before moving them to a server.

🔹 FTP Task

✅ **Purpose:** Uploads/downloads files to/from an **FTP server**.

✅ **When to Use:**

- Retrieving external data files from an FTP server.
- Sending processed files to a remote location.

📌 **Example:** Download daily sales data from an FTP server for processing.

🔹 Web Service Task

✅ **Purpose:** Calls a **web service** and retrieves data.

✅ **When to Use:**

- Fetching real-time data (e.g., currency exchange rates, weather data).
- Integrating with external applications that expose web services.

📌 **Example:** Get stock prices from a financial web service before analysis.

🔹 Execute Package Task

✅ **Purpose:** Runs another SSIS package within the current package.

✅ **When to Use:**

- Breaking down large workflows into smaller, reusable packages.
- Managing dependencies between different ETL processes.

📌 **Example:** A master package executes individual packages for **Extract, Transform, and Load (ETL)** separately.

🔹 Analysis Services Processing Task

✅ **Purpose:** Processes **SSAS cubes, dimensions, or partitions**.

✅ **When to Use:**

- Refreshing SSAS cubes with the latest data after ETL completes.

📌 **Example:** After loading new sales data, process the SSAS cube to update reports.

🔹 Data Profiling Task

✅ **Purpose:** Analyzes data quality by profiling a dataset.

✅ **When to Use:**

- Checking for missing or duplicate values before loading data.
- Validating that incoming data meets business rules.

📌 **Example:** Check if customer email addresses are missing before importing data.

2. Using Tasks in Control Flow

🔹 How Control Flow Works in SSIS?

- **Tasks are executed sequentially or in parallel** based on **precedence constraints** (arrows between tasks).
- **Success (Green Line)** → Next task runs if the previous task succeeds.
- **Failure (Red Line)** → Next task runs only if the previous task fails.

- **Completion (Blue Line)** → Next task runs regardless of success or failure.

📌 **Example Workflow in Control Flow:**
1️⃣ **Execute SQL Task** → Truncate staging table.
2️⃣ **Data Flow Task** → Load new data from CSV to the staging table.
3️⃣ **Execute SQL Task** → Run stored procedure to update the main table.
4️⃣ **File System Task** → Move the processed file to an archive folder.
5️⃣ **Send Mail Task** → Send an email notification if the process succeeds.

3. Choosing the Right Task for Your SSIS Package

Task Type	Purpose	When to Use
Data Flow Task	Moves & transforms data	ETL (Extract, Transform, Load)
Execute SQL Task	Runs SQL commands	Running queries or stored procedures
File System Task	Manages files/folders	Moving, copying, deleting files
Script Task	Runs custom code	Complex logic, API calls, automation
Send Mail Task	Sends emails	Notifications on success/failure
Execute Process Task	Runs external apps	Calling PowerShell, batch files
FTP Task	Transfers files via FTP	Download/upload external data
Web Service Task	Calls web services	Fetching live data from APIs
Execute Package Task	Runs another SSIS package	Modularizing large ETL processes
Analysis Services Processing Task	Refreshes SSAS cubes	Updating data models
Data Profiling Task	Checks data quality	Validating missing/duplicate values

4. Best Practices for SSIS Tasks & Control Flow

✔ **Use Precedence Constraints:** Control execution based on success, failure, or completion.
✔ **Optimize Data Flow Tasks:** Minimize unnecessary transformations for better performance.
✔ **Use Execute SQL Task Efficiently:** Prefer stored procedures over inline queries for reusability.
✔ **Modularize with Execute Package Task:** Break complex workflows into smaller packages.

✔ **Handle Errors with Logging & Email Alerts:** Capture failures with event handlers and send notifications.
✔ **Avoid Excessive Script Tasks:** Use built-in SSIS tasks when possible to maintain performance.

Configuring the Execute SQL Task in SSIS

The **Execute SQL Task** allows you to run **SQL queries or stored procedures** within an SSIS package. It's commonly used for:
✅ Creating, updating, or deleting database records
✅ Running stored procedures
✅ Truncating tables before loading data
✅ Creating tables dynamically

Configuring Execute SQL Task in SSIS

1. Add the Task

- Drag **Execute SQL Task** from SSIS Toolbox to **Control Flow**.

2. Configure the Task

- **Double-click** the task → Set **Connection** (OLE DB, ADO.NET).
- Enter **SQL Statement** (e.g., TRUNCATE TABLE Sales_Staging;).
- **ResultSet**:
  - **None** → No output (e.g., INSERT, DELETE).
  - **Single Row** → If expecting **one value** (SELECT COUNT(*)).
  - **Full Result Set** → For multiple rows.

3. Handle Parameters (Optional)

- Use **?** for **OLE DB**, named parameters for **ADO.NET**.
- Map **SSIS Variables** in **Parameter Mapping**.
  - Example: DELETE FROM Orders WHERE OrderDate < ?;
  - Parameter: User::OrderDate, Data Type: DATE.

4. Test & Run

- Click **OK**, then **Execute Task**.
- Check **Progress** tab for errors.

Best Practices

✔ Use **parameterized queries**.
✔ Optimize SQL queries (use stored procedures).
✔ Handle **errors** with event handlers.
🔷 **Next: File System Task or Data Flow Task?** 🚀

Configuring the Backup Database Task

✅ **Connection**: Select or create an **OLE DB connection** to SQL Server.
✅ **Backup Type**: Choose **Full, Differential, or Transaction Log**.
✅ **Databases**:

- Select **one or multiple** databases.
- Use All databases for a full instance backup.
  ✅ **Backup Destination**:
- Select **Disk** or **Tape**.
- Provide the **backup file location** (e.g., C:\Backups\MyDB.bak).
  ✅ **Options**:
- **Append or Overwrite**: Choose whether to overwrite existing backups.

Steps to Configure Script Task in SSIS

1️⃣ **Drag Script Task** into **Control Flow**.
2️⃣ **Double-click** to open the editor.
3️⃣ **Choose Script Language** → **C#** or **VB.NET**.
4️⃣ **Select Variables** (ReadOnly/ReadWrite) if needed.
5️⃣ Click **Edit Script** → Write custom **C#** or **VB.NET** code.
6️⃣ Click **Save & Close**, then **OK**.
7️⃣ **Execute Task** and check the result.

Steps to Configure Bulk Insert Task in SSIS

1️⃣ **Drag Bulk Insert Task** into **Control Flow**.
2️⃣ **Double-click** to open the editor.
3️⃣ **Set Connection** → Select **OLE DB connection** (Destination DB).
4️⃣ **Specify Destination Table** (where data will be inserted).
5️⃣ **Set File Connection** → Choose **Flat File Source** (CSV, TXT).
6️⃣ **Configure Format** → Select **Delimiter, Row Terminator** if needed.
7️⃣ **Adjust Options** (e.g., **First Row**, Keep Nulls, Fire Triggers).
8️⃣ **Click OK** → **Execute Task** → Verify data in the table.
🔷 **Next: Data Flow Task?** 🚀

For Loop Container in SSIS

Use it when you need to **execute tasks multiple times** based on a **loop condition** (e.g., iterating over a counter, processing records in batches).

Steps to Configure

1️⃣ **Drag For Loop Container** into **Control Flow**.
2️⃣ **Double-click** to open the editor.
3️⃣ **Define Loop Conditions:**

- **InitExpression** → Set a variable (@Counter = 1).
- **EvalExpression** → Condition to continue (@Counter <= 10).

- **AssignExpression** → Increment (@Counter = @Counter + 1).

4 **Place Tasks Inside the Loop** (e.g., **Execute SQL Task**, **File Processing**).

5 Click → **Execute Package**.

🔷 **Next: Foreach Loop Container?** 🚀

Using Variables in SSIS and Passing to Script Task

Use when you need to store values dynamically and process them in a **Script Task**.

Steps to Configure

1. Create a Variable

1 Open **SSIS Package** → Go to **SSIS Variables** pane.

2 Click **Add Variable** → Set Name (User::MyVariable).

3 Choose **Data Type** (e.g., Int32, String).

4 Set an **Initial Value** (e.g., "Hello, SSIS!").

2. Add Script Task and Pass Variable

5 Drag **Script Task** into **Control Flow**.

6 **Double-click** → Click **Edit Script**.

7 In **ReadOnlyVariables**, add User::MyVariable.

8 Click **Edit Script** → Inside Main() method, add:

```
using System;
using System.Windows.Forms;
using Microsoft.SqlServer.Dts.Runtime;
public void Main()
{
   string myValue = Dts.Variables["User::MyVariable"].Value.ToString();
   MessageBox.Show("Variable Value: " + myValue);
   Dts.TaskResult = (int)ScriptResults.Success;
}
```

9 **Save & Close** → **Execute Package**.

🔷 **Next: Foreach Loop with Variables?** 🚀

Dts (Data Transformation Services) in SSIS refers to the **Data Transformation Service Package** API, which allows you to interact with SSIS objects within a **Script Task**.

Foreach Loop Container in SSIS

Use when you need to **iterate over a collection** (e.g., files in a folder, rows in a dataset, items in an array).

Steps to Configure

1 **Drag Foreach Loop Container** into **Control Flow**.

2 **Double-click** to open the editor.

3️⃣ **Set Enumerator Type** based on the use case:
- **Foreach File Enumerator** → Loop through files in a folder.
- **Foreach ADO Enumerator** → Loop through dataset rows.
- **Foreach Item Enumerator** → Loop through hardcoded values.

4️⃣ **Configure Enumerator Properties** (e.g., select folder path for files).
☐ **Create an SSIS Variable** (User::FileName) to store the loop value.
6️⃣ **Go to Variable Mappings** → Assign the variable to the loop output.
☐ **Place Tasks Inside the Loop** (e.g., **Execute SQL Task, Script Task)**.
8️⃣ Click **OK** → **Execute Package**.
🔷 **Next: Need to loop through files or database records?** 🚀

Set Connection Manager String Dynamically in SSIS Using a Variable

Use when you need to dynamically set the file path or database connection string.

Steps to Configure

1️⃣ **Create an SSIS Variable**
- Go to **SSIS Variables** pane.
- Click **Add Variable** → Name it (User::MyPath).
- Set **Data Type** → String.
- Set **Value** → e.g., "C:\Data\myfile.csv" (for a file) or "Server=MyServer;Database=MyDB;Integrated Security=True;" (for a database).

2️⃣ **Assign Variable to Connection Manager**
- Right-click **Connection Manager** (e.g., Flat File, OLE DB).
- Click **Properties** → Find **Expressions**.
- Click **[...]** next to Expressions.
- Select **ConnectionString** → Set @[User::MyPath].
- Click **OK**.

3️⃣ **Execute Package** → The connection will now use the variable dynamically.
🔷 **Next: Need to update it inside a Script Task?** 🚀

Data Flow Task in SSIS

Use when you need to **extract, transform, and load (ETL) data** between sources and destinations.

Steps to Configure

1️⃣ **Drag Data Flow Task** into **Control Flow**.
2️⃣ **Double-click** to enter the **Data Flow** tab.
3️⃣ **Add a Source Component**:
- Drag **Flat File Source**, **OLE DB Source**, or another source type.
- Configure **Connection Manager** (select file, table, or query).

4️⃣ **(Optional) Add Transformations**:
- Use **Derived Column** to create new columns.
- Use **Data Conversion** to change data types.

- Use **Conditional Split** to filter rows.

5 **Add a Destination Component**:
- Drag **OLE DB Destination**, **Flat File Destination**, etc.
- Map source columns to **destination columns**.

6 Click **OK** → **Execute Package** to load data.

🔷 **Next: Need to handle errors in Data Flow?** 🚀

OLE DB Source in SSIS

Use when you need to extract data from a **relational database** (SQL Server, Oracle, etc.).

Steps to Configure

1 **Drag Data Flow Task** into **Control Flow**.
2 **Double-click** to enter the **Data Flow** tab.
3 **Drag OLE DB Source** into the Data Flow.
4 **Double-click OLE DB Source** → Configure:
- **OLE DB Connection Manager** → Select or create a connection.
- **Data Access Mode**:
  - **Table or View** → Select a table.
  - **SQL Command** → Write a custom query.
- **Preview Data** → Verify output.

5 Click **OK** → Connect to the next component (Transformation or Destination).

☐**Execute Package** to extract data.

🔷 **Next: OLE DB Destination?** 🚀

Excel Destination in SSIS

Use when you need to export data from SSIS to an **Excel file**.

Steps to Configure

1 **Drag Data Flow Task** into **Control Flow**.
2 **Double-click** to enter the **Data Flow** tab.
3 **Add a Source Component** (e.g., **OLE DB Source** for database data).
4 **Drag Excel Destination** into the Data Flow.
5 **Connect Source to Excel Destination**.
6 **Double-click Excel Destination** → Configure:
- **Excel Connection Manager** → Select/Create Excel file (.xlsx).
- **Table or Sheet Name** → Select or create a sheet.
- **Mappings** → Ensure source columns match destination columns.

7 Click **OK** → **Execute Package** to export data.

🔷 **Next: Need dynamic file paths for Excel?** 🚀

If there's a conflict between **SQL Server** and **Excel** (usually because Excel drivers are **32-bit only**), you can fix it by disabling **Run64BitRuntime** in SSIS.

Steps to Fix Excel & SQL Service Conflict in SSIS

1️⃣ Open **SSIS Project** in **Visual Studio**.
2️⃣ Go to **Solution Explorer** → Right-click **Project** → Click **Properties**.
3️⃣ Navigate to **Configuration Properties** → **Debugging**.
4️⃣ Find **Run64BitRuntime** → Set it to **False**.
5️⃣ Click **Apply** → **OK**.
6️⃣ **Execute the Package** again.
🔷 **Next: Need to handle Excel file dynamically?** 🚀

Excel file to Data warehouse Dimension

Character Map Transformation in SSIS

Use when you need to modify text data by changing **case (uppercase/lowercase), converting symbols, or handling special characters**.

Steps to Configure

1️⃣ **Drag Data Flow Task** into **Control Flow**.
2️⃣ **Double-click** to enter the **Data Flow** tab.
3️⃣ **Add a Source Component** (e.g., **OLE DB Source**).
4️⃣ **Drag Character Map Transformation** into the Data Flow.
5️⃣ **Connect Source to Character Map Transformation**.
6️⃣ **Double-click Character Map** → Select the **column(s)** to modify.
7️⃣ Choose **Transformation Type** (e.g., UPPERCASE, lowercase, byte reversal).
8️⃣ Set **New Column Name** or **Overwrite Existing Column**.
9️⃣ Click **OK** → Connect to **Destination Component** (e.g., OLE DB Destination).
🔟 **Execute Package** to apply character transformations.

Derived Column Transformation in SSIS

Use when you need to create **new calculated columns** or **modify existing data** in a Data Flow.

Steps to Configure

1️⃣ **Drag Data Flow Task** into **Control Flow**.
2️⃣ **Double-click** to enter the **Data Flow** tab.
3️⃣ **Add a Source Component** (e.g., **OLE DB Source**).
4️⃣ **Drag Derived Column Transformation** into the Data Flow.
5️⃣ **Connect Source to Derived Column**.
6️⃣ **Double-click Derived Column** → Configure:

- Select a **column** to modify or **create a new column**.

- Use **SSIS expressions** for transformations (e.g., calculations, string manipulation).

Examples of Derived Column Expressions

- **Convert to UPPERCASE:**
  UPPER(FirstName)
- **Concatenate columns:**
  FirstName + " " + LastName
- **Replace NULL with default value:**
  ISNULL(Email) ? "NoEmail@domain.com" : Email
- **Calculate Age (assuming DOB column):**
  DATEDIFF("Year", DOB, GETDATE())

7 Click **OK** → Connect to **Destination Component** (e.g., OLE DB Destination).
8 **Execute Package** to apply transformations.

Copy Column Transformation in SSIS

Use when you need to **duplicate a column** while keeping the original data unchanged (e.g., for modifications using other transformations).

Steps to Configure

1 **Drag Data Flow Task** into **Control Flow**.
2 **Double-click** to enter the **Data Flow** tab.
3 **Add a Source Component** (e.g., **OLE DB Source**).
4 **Drag Copy Column Transformation** into the Data Flow.
5 **Connect Source to Copy Column**.
6 **Double-click Copy Column** → Select column(s) to copy.
7 **Rename copied column(s)** if needed (e.g., OriginalColumn_Copy).
8 Click **OK** → Connect to another transformation (e.g., **Derived Column** or **Character Map**).
9 **Execute Package** to apply changes.

Data Conversion Transformation in SSIS

Use when you need to **change data types** (e.g., convert string to integer, DT_DATE to DT_STR).

Steps to Configure

1 **Drag Data Flow Task** into **Control Flow**.
2 **Double-click** to enter the **Data Flow** tab.
3 **Add a Source Component** (e.g., **OLE DB Source**).
4 **Drag Data Conversion Transformation** into the Data Flow.
5 **Connect Source to Data Conversion**.
6 **Double-click Data Conversion** → Configure:

- Select column(s) to convert.
- Set **New Column Name**.

- Choose **Data Type** (e.g., DT_I4 for Integer, DT_STR for String).
- Set **Length, Precision, Scale** if needed.
  7️⃣ Click **OK** → Connect to Destination (e.g., **OLE DB Destination**).
  ☐**Execute Package** to apply conversions.

Sort Transformation in SSIS

Use when you need to **sort data** in ascending or descending order before loading into a destination or performing lookups.

Steps to Configure

1️⃣ **Drag Data Flow Task** into **Control Flow**.
2️⃣ **Double-click** to enter the **Data Flow** tab.
3️⃣ **Add a Source Component** (e.g., **OLE DB Source**).
4️⃣ **Drag Sort Transformation** into the Data Flow.
5️⃣ **Connect Source to Sort Transformation**.
6️⃣ **Double-click Sort Transformation** → Configure:

- Check the column(s) you want to **sort by**.
- Choose **Ascending** or **Descending** order.
- (Optional) **Remove duplicates** by checking **"Remove rows with duplicate sort values"**.
  7️⃣ Click **OK** → Connect to Destination (e.g., **OLE DB Destination**).
  ☐**Execute Package** to apply sorting.

Audit Transformation in SSIS

Use when you need to **track metadata** about the package execution (e.g., machine name, execution time, user, package ID).

Steps to Configure

1️⃣ **Drag Data Flow Task** into **Control Flow**.
2️⃣ **Double-click** to enter the **Data Flow** tab.
3️⃣ **Add a Source Component** (e.g., **OLE DB Source**).
4️⃣ **Drag Audit Transformation** into the Data Flow.
5️⃣ **Connect Source to Audit Transformation**.
6️⃣ **Double-click Audit Transformation** → Configure:

- Select the **Audit Type** (e.g., ExecutionInstanceGUID, PackageID, UserName).
- Rename the **Output Column Name** if needed.
  7️⃣ Click **OK** → Connect to Destination (e.g., **OLE DB Destination**).
  ☐**Execute Package** to capture metadata.

Multicast Transformation in SSIS

Use when you need to **send the same data to multiple destinations** for parallel

processing.

Steps to Configure

1. **Drag Data Flow Task** into **Control Flow**.
2. **Double-click** to enter the **Data Flow** tab.
3. **Add a Source Component** (e.g., **OLE DB Source**).
4. **Drag Multicast Transformation** into the Data Flow.
5. **Connect Source to Multicast Transformation**.
6. **Drag multiple Destination Components** (e.g., **OLE DB Destination**, **Flat File Destination**, **Excel Destination**).
7. **Connect Multicast Output to Each Destination**.
8. **Configure Each Destination Separately**.
9. **Execute Package** to distribute data across multiple destinations.

Conditional Split Transformation in SSIS

Use when you need to **route data into multiple outputs** based on conditions (e.g., filter sales by region, separate valid and invalid data).

Steps to Configure

1. **Drag Data Flow Task** into **Control Flow**.
2. **Double-click** to enter the **Data Flow** tab.
3. **Add a Source Component** (e.g., **OLE DB Source**).
4. **Drag Conditional Split Transformation** into the Data Flow.
5. **Connect Source to Conditional Split Transformation**.
6. **Double-click Conditional Split** → Configure:
   - **Define conditions** using SSIS expressions.
   - Example conditions:
     - **Sales > 1000 → "High Sales"**
       SalesAmount > 1000
     - **Region = 'USA' → "US Sales"**
       Region == "USA"
     - **Default output for unmatched data**.
       7. **Click OK** → Connect Each Output to a Destination (e.g., OLE DB Destination).
       ☐**Execute Package** to apply data routing.

Union All Transformation in SSIS

Use when you need to **combine multiple data sources into a single output** (similar to SQL UNION ALL, but without removing duplicates).

Steps to Configure

1. **Drag Data Flow Task** into **Control Flow**.
2. **Double-click** to enter the **Data Flow** tab.
3. **Add multiple Source Components** (e.g., **OLE DB Source**, **Flat File Source**).

4 **Drag Union All Transformation** into the Data Flow.
5 **Connect each Source to Union All Transformation**.
6 **Double-click Union All** → Ensure **column mappings are correct** (data types must match).
7 Click **OK** → Connect the Union All output to a Destination (e.g., **OLE DB Destination**).
8 **Execute Package** to merge the data.

Merge Join Transformation in SSIS

Use when you need to **combine data from two sorted sources** based on a common key (similar to SQL JOIN).

Steps to Configure

1 **Drag Data Flow Task** into **Control Flow**.
2 **Double-click** to enter the **Data Flow** tab.
3 **Add two Source Components** (e.g., **OLE DB Source**, **Flat File Source**).
4 **Sort both sources** using **Sort Transformation**:

- **Enable sorting** on the **join key column** in both sources.

5 **Drag Merge Join Transformation** into the Data Flow.
☐ **Connect both sorted sources** to Merge Join Transformation.
☑ **Double-click Merge Join** → Configure:

- Select **Join Type** (INNER JOIN, LEFT OUTER JOIN, FULL OUTER JOIN).
- Ensure **correct column mapping** for the join key.
- Choose **columns to include** in the output.

8 Click **OK** → Connect the output to a **Destination Component** (e.g., OLE DB Destination).
☐ **Execute Package** to join the data.
🔷 **Next: Need to merge unsorted data? Use Merge Transformation!** 🚀

Merge Transformation in SSIS

Use when you need to **combine two sorted datasets with the same structure** into a single output (similar to SQL UNION, but keeps sorting).

Steps to Configure

1 **Drag Data Flow Task** into **Control Flow**.
2 **Double-click** to enter the **Data Flow** tab.
3 **Add two Source Components** (e.g., **OLE DB Source**, **Flat File Source**).
4 **Sort both sources** using **Sort Transformation**:

- Enable sorting on the **merge key column** in both sources.

5 **Drag Merge Transformation** into the Data Flow.
☐ **Connect both sorted sources** to Merge Transformation.
☑ **Double-click Merge** → Ensure **column mappings match** (same data types).

8 Click → Connect the output to **a Destination Component** (e.g., OLE DB Destination) → **Execute Package** to merge the data.

Enable Data Viewer in SSIS

Use when you need to **inspect data flow between transformations** for debugging.

Steps to Configure:

1. **Open Data Flow Task** in SSIS.
2. **Right-click the Data Flow Path (blue arrow)** between two components.
3. Click **"Enable Data Viewer"**.
4. A **magnifying glass icon** appears on the path, indicating the Data Viewer is enabled.
5. **Run the package** → A pop-up window will display the data at that point.
6. Click **"Detach"** to continue execution or **"Stop Debugging"** to analyze further.