



Implementing ETL with SQL Server Integration Services

[OnlineUA] DWBI\DQE Program 2020

Andrii Pavlish

FOR YOUR INFORMATION

- Please turn off the microphone.
- If you have questions, ask them in the chat.
- Duration: 3 hours
- Coffee break 15 minute



Agenda

1. ETL PROCESSING
2. ETL PROCESSING WITH SSIS
3. SSIS DATA FLOWS
4. DEPLOYMENT AND TROUBLESHOOTING



WHAT YOU WILL LEARN

- Creating an ETL script
- The design environment
- Control flows
- Data sources
- Data transformations
- Data destinations
- Precedence constraints
- Connection managers
- Execute SQL tasks
- Progress/execution results
- Data flows
- Data flow paths
- Error output paths
- Configuring data sources and destinations
- Executing SSIS packages
- Deploying SSIS packages

Setting Up Your Environment

SQL SERVER DATA ENGINE(FREE DEVELOPER EDITION): [HTTPS://WWW.MICROSOFT.COM/EN-US/SQL-SERVER/SQL-SERVER-DOWNLOADS](https://www.microsoft.com/en-us/sql-server/sql-server-downloads)

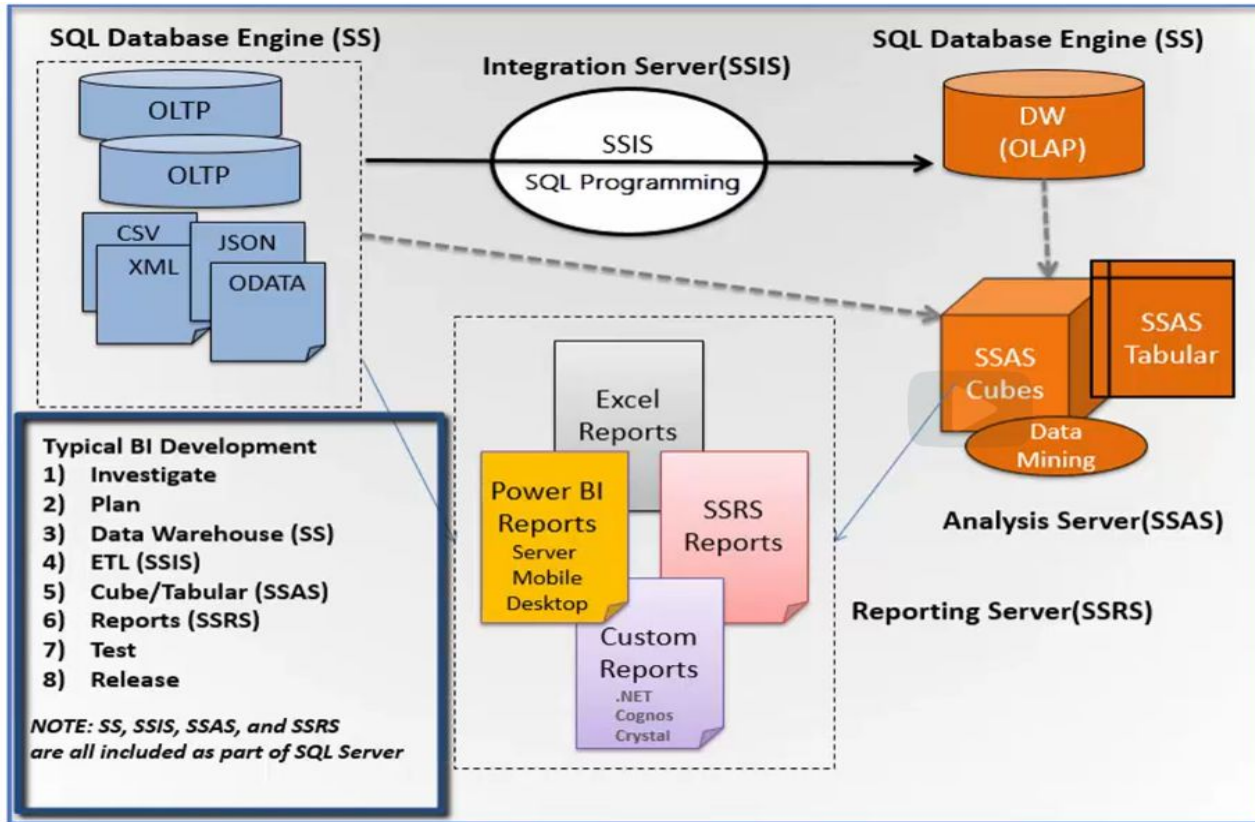
SQL SERVER MANAGEMENT STUDIO(SSMS 18.8) :

[HTTPS://DOCS.MICROSOFT.COM/EN-US/SQL/SSMS/DOWNLOAD-SQL-SERVER-MANAGEMENT-STUDIO-SSMS?VIEW=SQL-SERVER-2017](https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017)

SQL SERVER DATA TOOLS(SSDT FOR VISUAL STUDIO (VS) 2017) : [HTTPS://GO.MICROSOFT.COM/FWLINK/?LINKID=2124319](https://go.microsoft.com/fwlink/?linkid=2124319)

1. ETL PROCESSING

ETL process in typical BI Solution

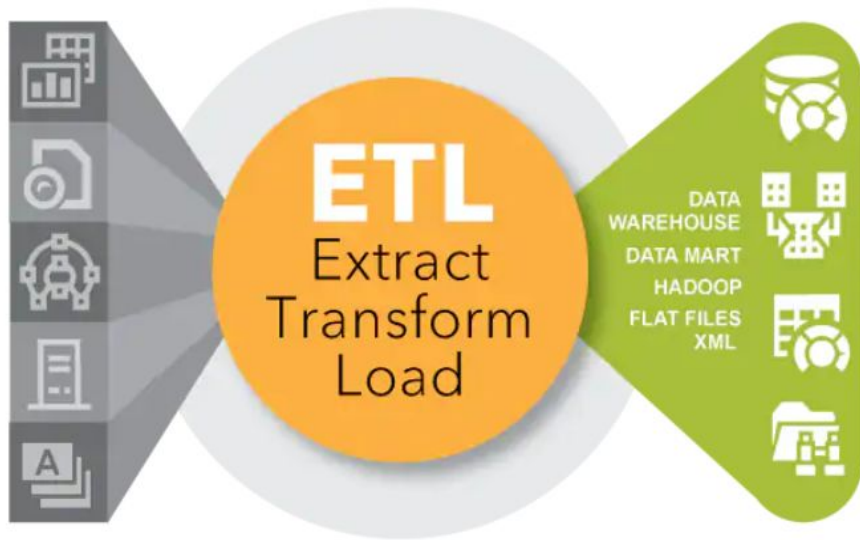


What is ETL?

ETL is a type of data integration that refers to the three steps (extract, transform, load) used to blend data from multiple sources. It's often used to build a data warehouse. During this process, data is taken (extracted) from a source system, converted (transformed) into a format that can be analyzed, and stored (loaded) into a data warehouse or other system. Extract, load, transform (ELT) is an alternate but related approach designed to push processing down to the database for improved performance.

Why ETL is Important

Businesses have relied on the ETL process for many years to get a consolidated view of the data that drives better business decisions. Today, this method of integrating data from multiple systems and sources is still a core component of an organization's data integration toolbox.



- When used with an enterprise data warehouse (data at rest), ETL provides deep historical context for the business.
- By providing a consolidated view, ETL makes it easier for business users to analyze and report on data relevant to their initiatives.
- ETL can improve data professionals' productivity because it codifies and reuses processes that move data without requiring technical skills to write code or scripts.
- ETL has evolved over time to support emerging integration requirements for things like streaming data. Organizations need both ETL and ELT to bring data together, maintain accuracy and provide the auditing typically required for data warehousing, reporting and analytics.

DEMO 1.

ETL with SCRIPTING

ETL Tools

- SQL Server Management Studio

SSMS is not designed specifically as a ETL processing application, however, it is still a great choice for this purpose. As shown earlier, BI professional can create and test transformation code within SSMS. Once this code is tested it can then be encapsulated into **views** and **stored procedures** which save the code within the database. From SSMS, you can also access and configure automations using SQL Server Agent.

- Visual Studio

Visual Studio itself is only an application for hosting development tools. These tools, plug into visual Studio providing a custom development environment. SQL Server Integration Services (SSIS) and SQL Server Data Tools (SSDT) are the two most common developer tool installed for ETL processing. The SSDT also includes advanced development tools for programming SQL Server Integration Server (SSIS) ETL packages, SQL Server Analysis Server (SSAS) Cubes, and SQL Server Reporting Server (SSRS) reports

Data Sources and Destinations Files

In order for a SSIS package to perform ETL Processing, you must configure its data sources and destinations. Each source and destination needs a connection and there are different kinds of connections.

- text files.

They're common because they are easy to generate and can be used on most operating systems without additional software (CSV, XML, JSON)

- databases

Most database applications provide data validation, data constraints, mapped relationships between sets of data, tools for automating common tasks, programming constructs (like views and stored procedures), and ways to access and change the data from a dedicated GUI. Because of this, using a database to store data is considered a better choice in comparison to text files.

- Web Services

In many cases the purpose of a given service is to return text data when requested. This text data may then be stored in a local text file or imported into a database.

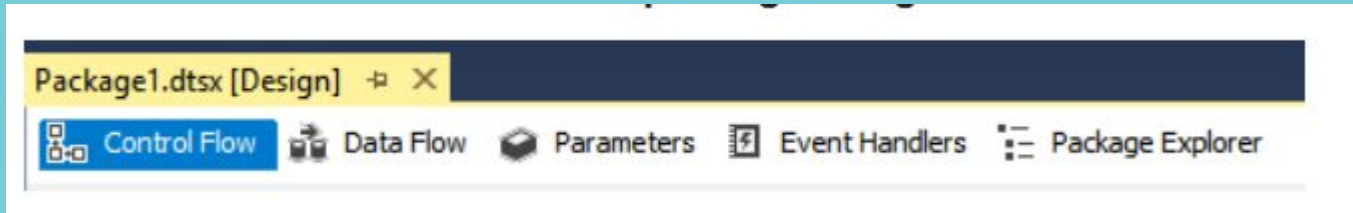
2. ETL PROCESSING WITH SSIS

DEMO 2.

SSIS OVERVIEW

Creating SSIS Project and Packages

- The Integration Services Project template uses one starter SSIS package that contains the programming instructions for your ETL process
- One or more SSIS packages can make up an SSIS project.
- SSIS packages are literally code files, and the code within an SSIS package is programmed using a designer user interface (UI).
- The designer is organized into 5 tabs.
- The Control Flow and the Data Flow tabs are used most often.



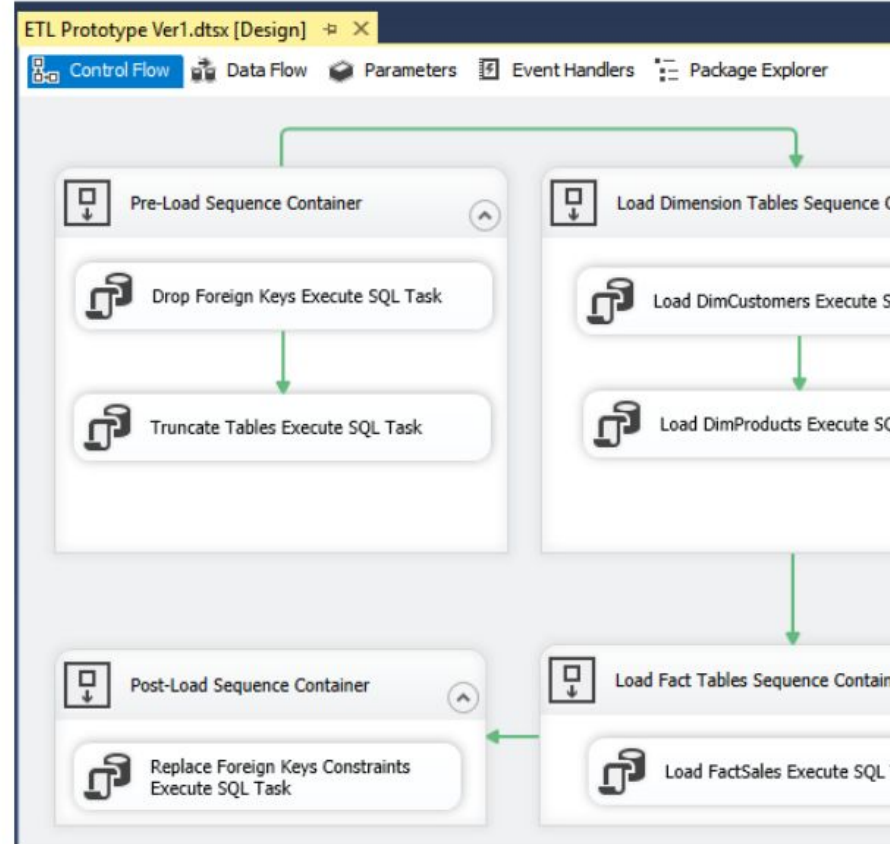
The Control Flow Tab

The control flow is created by dragging sequence containers and tasks from the SSIS toolbox onto the design surface. As the name implies, lets you control the flow of the package.

The most common control flow tasks are:

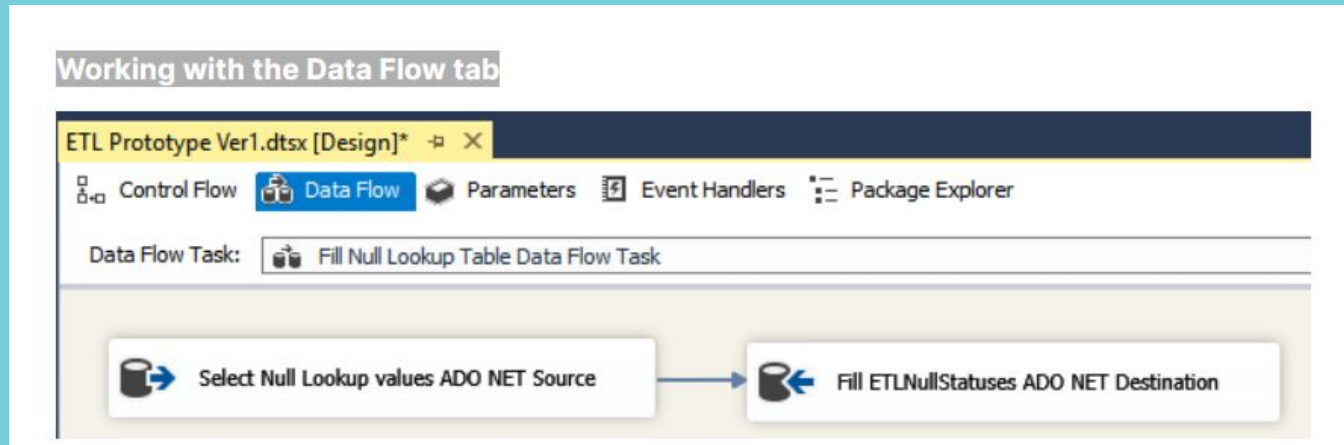
- *Annotations:* Text blocks that contain no data.
- *Data Flow Task:* Moves data between sources and destinations.
- *Execute SQL Task:* Runs the statement or statements on the data source.
- *Sequence container:* Groups tasks together.

Working with the control flow tab



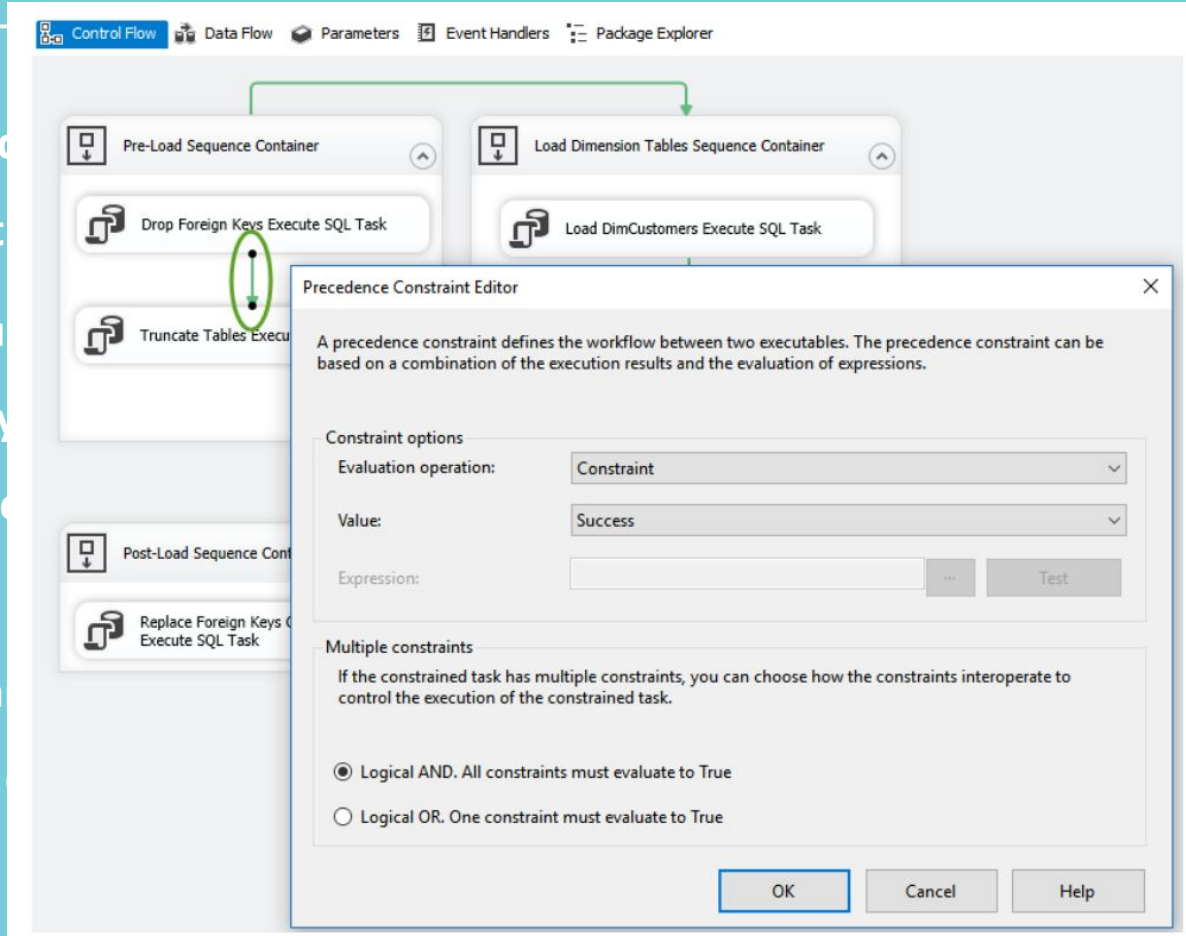
The Data Flow Tab

- Data flows are the only task that have their own tab.
- Data flow tasks encapsulate the data flow engine
- Are specialized for transferring data from one location to another.



Sequence Containers and Precedent Constraints

- Sequence Containers are used to group tasks (f.e. dimension tables, or fact tables).
- Naming conventions for sequence containers help to identify their purpose. Once you have named your containers, they can then be configured.
- The precedence constraints are used to define the logic such as success, failure,



SSIS Connections

Connections are added from the Connection Manager

The three most frequently used are:

- The OLE DB connection manager

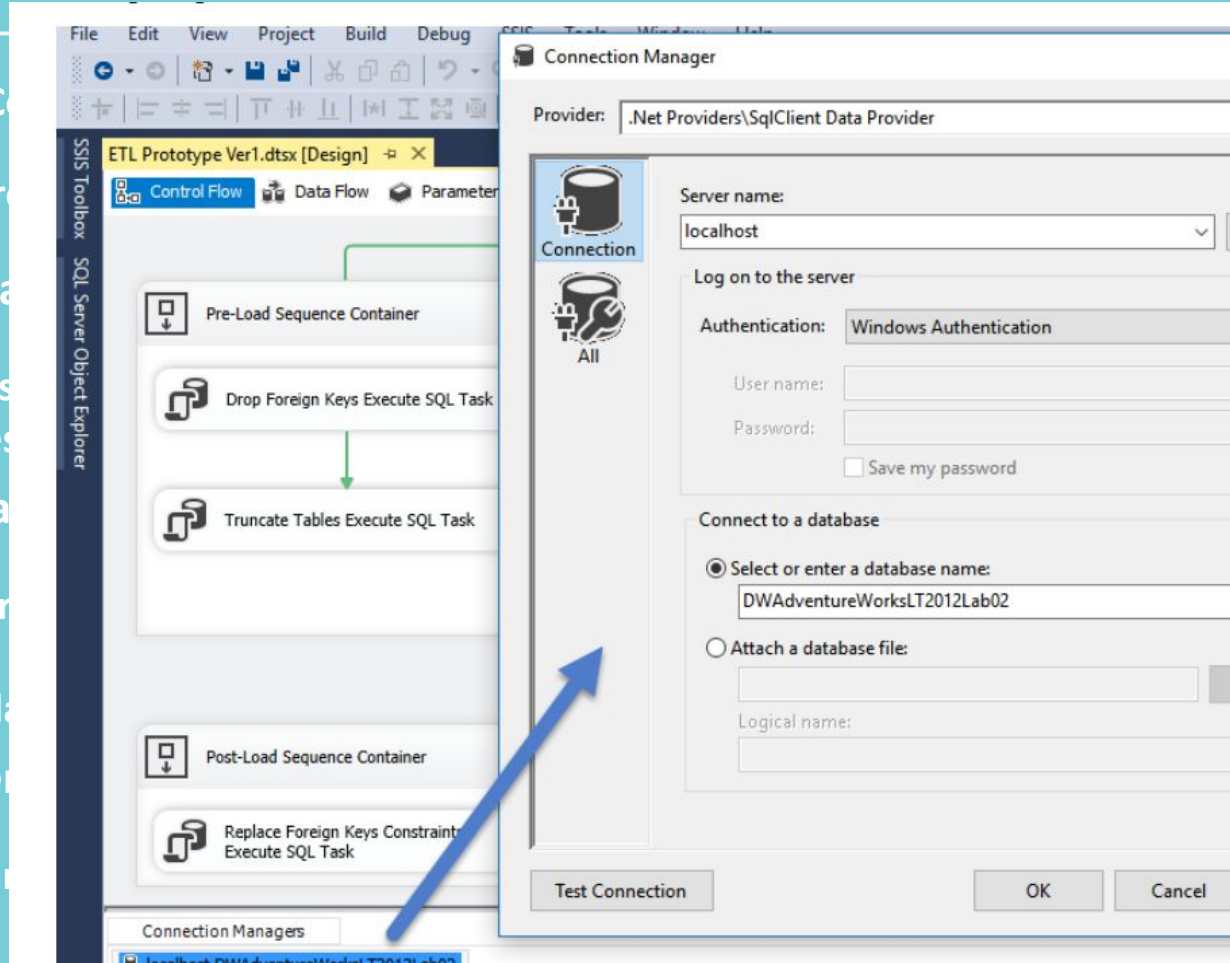
The OLE DB connection manager is easier than other connection types

- The ADO.NET connection manager

The ADO.NET connection manager features increased performance
Types are based on the .NET standard

- The File Connection Manager

The file connection manager can

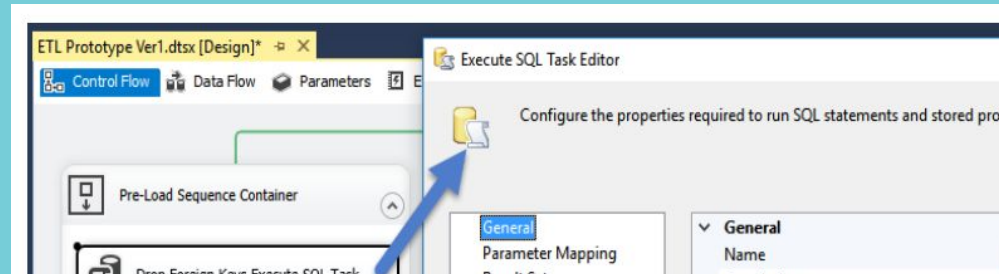


Configuring Execute SQL Tasks

Execute SQL task allows you to run SQL code or stored procedures from a package on a connected database. The task can run a single statement, or multiple sequential statements.

The Execute SQL tasks can be used for the following:

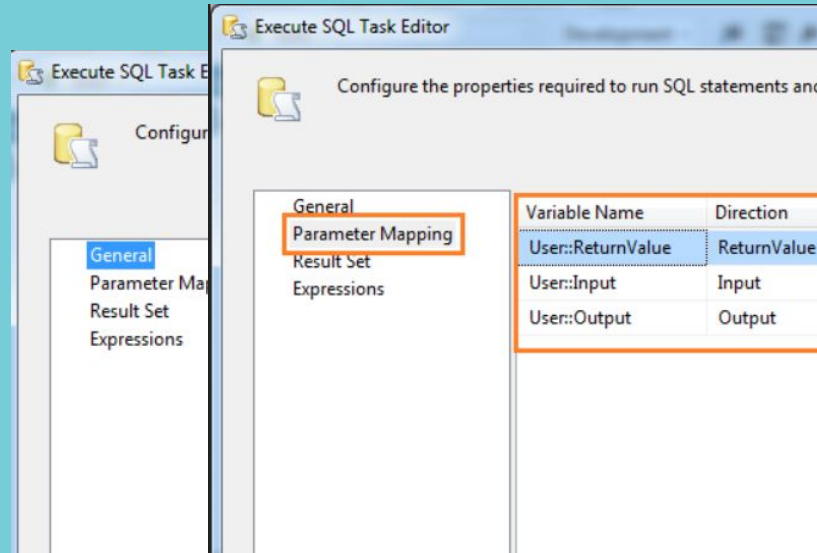
- Drop foreign key constraints
- Re-create fact and dimension tables
- Modify database tables and views by creating, altering, or dropping them
- Truncate a table's data
- Run stored procedures
- Save returned rowset objects into a variable



Using Stored Procedures from SSIS

When using stored procedures in SSIS you will need to consider the following:

- What types of connections you will you use?
- Does the stored procedure have parameters?
- Will the stored procedure return data?



DEMO 3.

Control flows and Data flows.

Containers and Precedence constraints

Connection Manager.

Execute stored procedure in SQL task

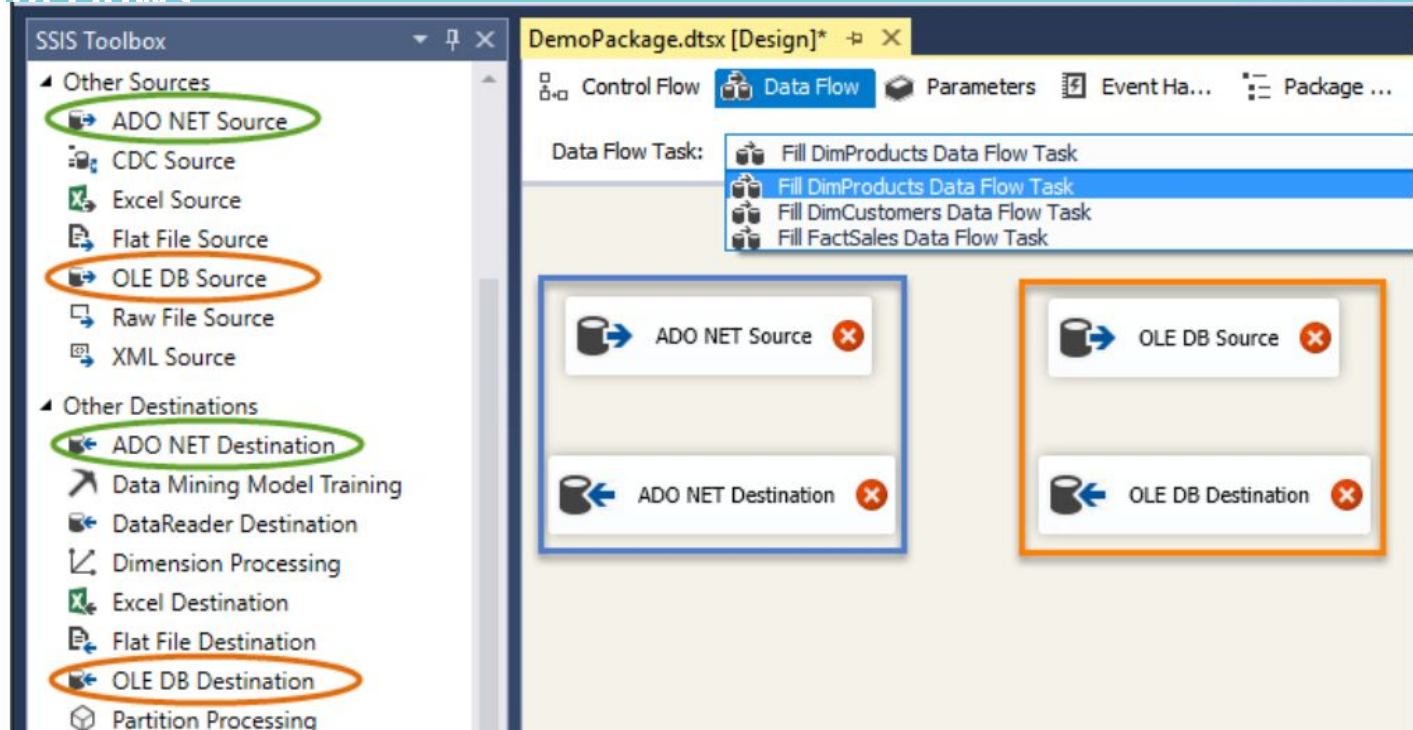
3. SSIS DATA FLOWS

Creating Data Flows

- Data flow tasks are made up of one or more components.
- Sources:
- Transform
- Destination

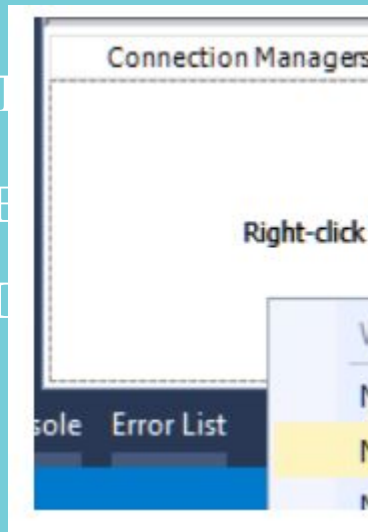
Data Flow component in your SSIS package. If you're using an OLE DB connection in your SSIS package, use a

corresponding OLE DB Source component.



s are

The OLE DB (Source) Connection Manager Page



The OLE DB Source Editor dialog box is shown, titled 'OLE DB Source Editor'. It contains the following fields and controls:

- Configure the properties used by a data flow to obtain data from any OLE DB provider.**
- Connection Manager:** A dropdown menu with a 'New...' button next to it.
- Data access mode:** A dropdown menu with 'Table or view' selected.
- Name of the table or the view:** A text box.

Data Access Mode

Data access

- Table
- Table
- adde
- SQL
- SQL
- proc

Selecting a Data Access Mode

Connection Manager

Columns

Error Output

Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.

OLE DB connection manager:

localhost.DWAdventureWorksLT2012v1OLEDB

New...

Data access mode:

Table or view

Table or view

Table name or view name variable

SQL command

SQL command from variable

Note: The table or view and table name or view name variable options bring all columns from the selected table. To restrict data necessary to your ETL process, we recommend using the SQL command or SQL command from variable Data access modes in combination with SQL programming statements.

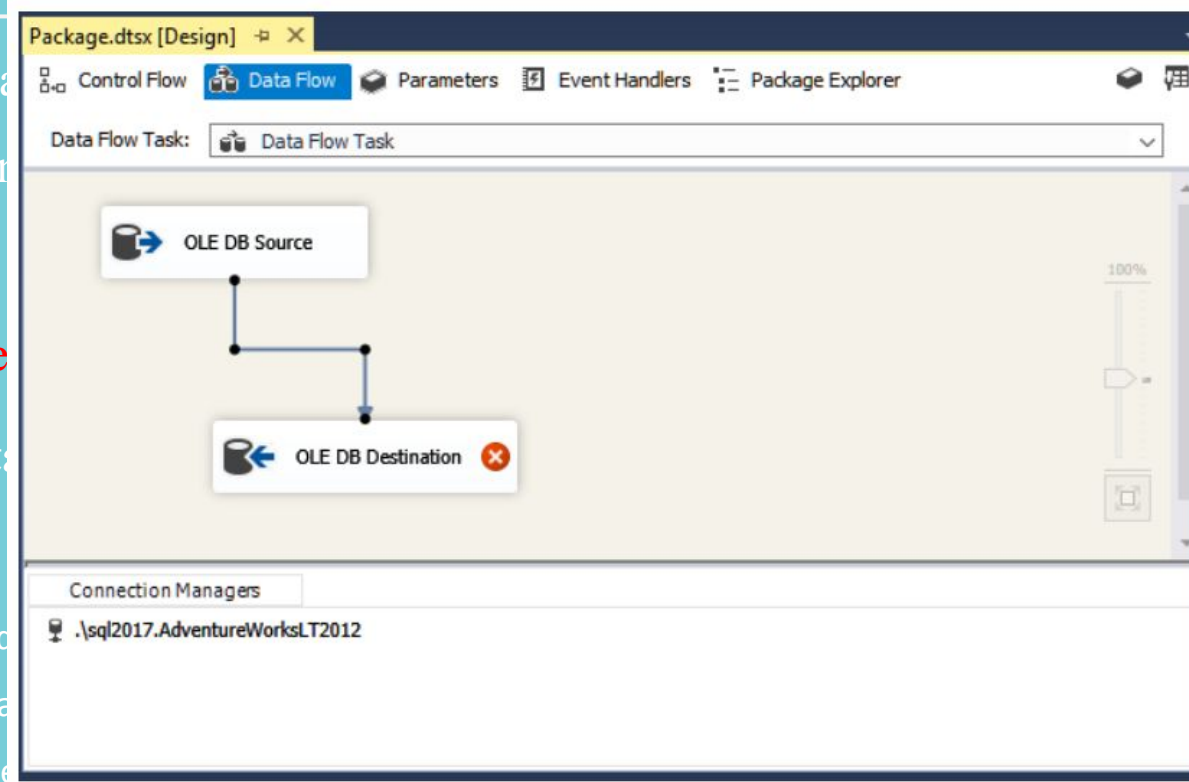
Data Flow Paths

- Data flow paths are represented as a sequence of components. A source component must be connected to a destination component (but not necessarily in that order).

Important: Be sure to connect the components correctly.

Configure the source data flow to the destination data flow arrows.

A **blue line** (representing the data flow), and allow for conditions to be configured, such as a transformation component onto the destination component.



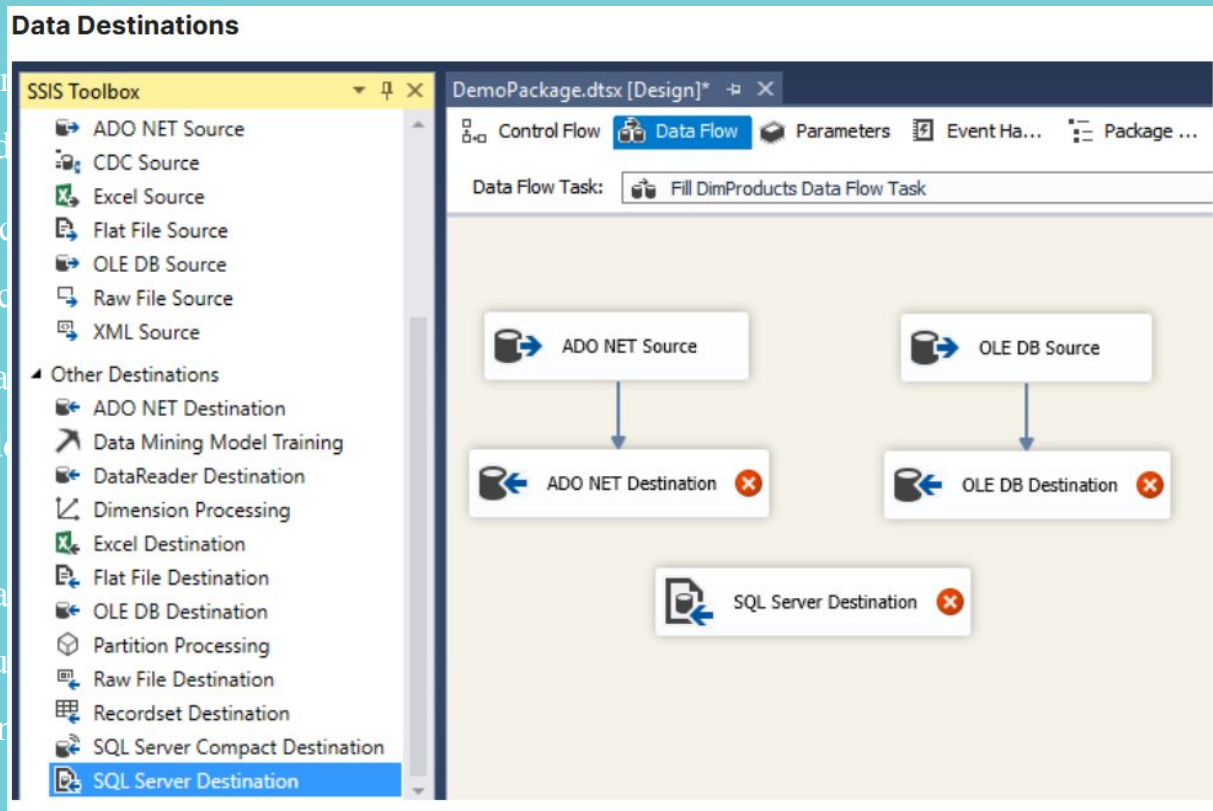
Data Destinations

You must have an un-configured data flow path has been added.

OLE DB destination (most common) is used to write data to a database table, view, or SQL database.

ADO.NET destinations are also used to write data to destinations. However, they do not support data type conversions.

For best performance, you may want to use a SQL Server database. When using a SQL Server database, you may be incurring data type conversion costs.



The (Destination) Connection Manager Page

On the Connection Manager page of the Destination Editor box to select an existing connection or use its New button to create a new connection.

Next, Use the Data access mode dropdown box to select one of the following options:

- **Table or view:** allows you to insert values into a new or existing table or view.
- **Table or view – fast mode:** allows you to bulk insert into a table or view. This mode provides additional configuration options and is easy to use.
- **Table name or view name variable:** allows you to use a variable to specify the table or view.
- **Table name or view name variable fast load:** allows you to use a variable to specify the table or view through an SSIS variable.
- **SQL command:** allows you to enter a SQL statement to insert data into a table or view.

The Destination Connection Manager Page

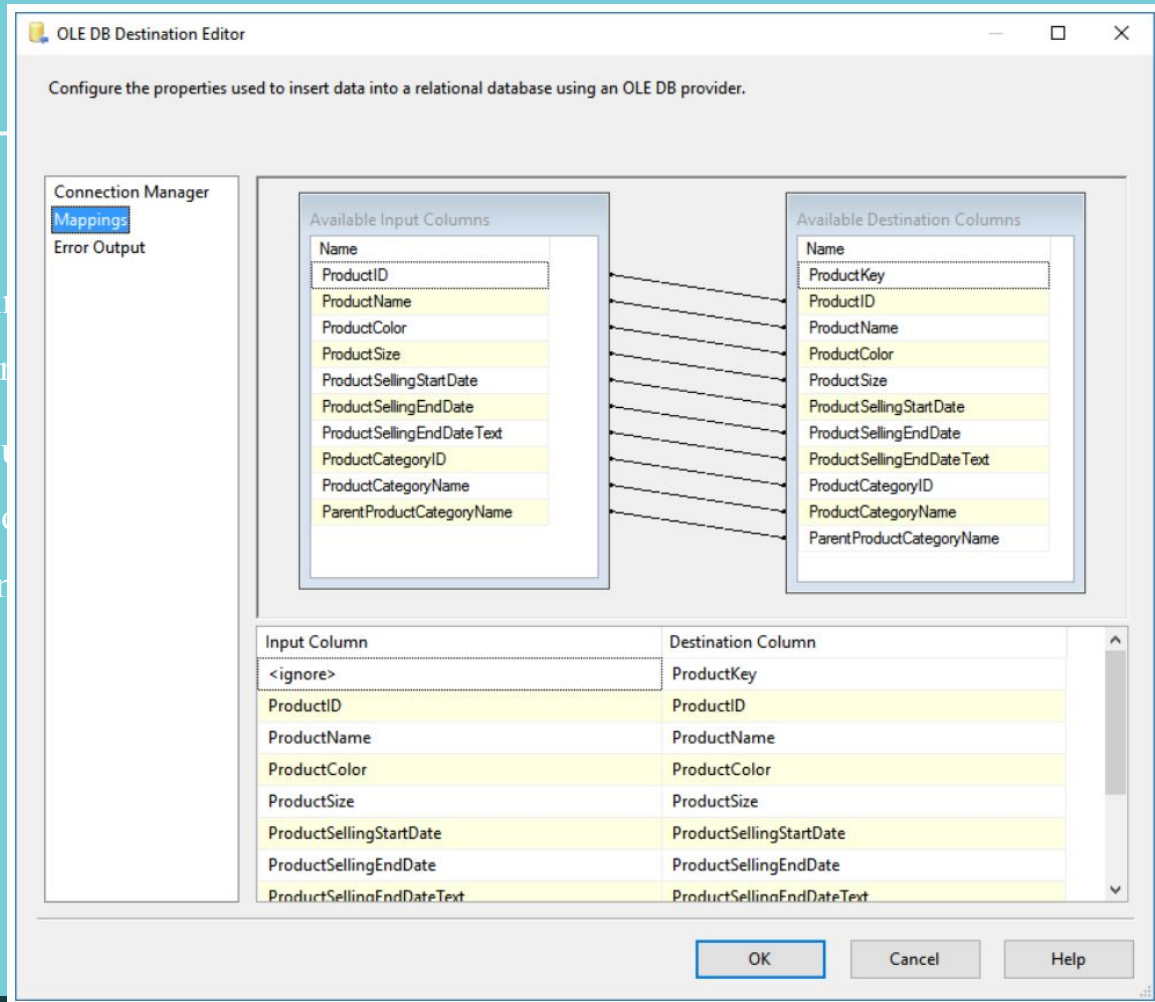
The screenshot shows the 'OLE DB Destination Editor' window. The title bar reads 'OLE DB Destination Editor'. Below the title bar, a subtitle says 'Configure the properties used to insert data into a relational database using an OLE DB provider.' On the left, there is a sidebar with three tabs: 'Connection Manager' (selected), 'Mappings', and 'Error Output'. The main area contains the following fields and controls:

- OLE DB connection manager:** A dropdown menu showing 'localhost.DWAAdventureWorksLT2012v1OLEDB' with a 'New...' button to its right.
- Data access mode:** A dropdown menu showing 'Table or view - fast load'.
- Name of the table or the view:** A dropdown menu showing '[dbo].[DimProducts]' with a 'New...' button to its right.
- Checkboxes:**
 - ☐ Keep identity
 - ☐ Keep nulls
 - ☒ Table lock
 - ☒ Check constraints
- Rows per batch:** A text box containing the value '1'.
- Maximum insert commit size:** A text box containing the value '2147483647'.
- View Existing Data...** button.

At the bottom of the window, there is a yellow warning bar with a warning icon and the text 'Map the columns on the Mappings page.' Below the warning bar are three buttons: 'OK', 'Cancel', and 'Help'.

The Mappings Page

- The Mappings page allows you to map input column names to output column names (when using a destination that requires column names to match). If they do not match, you can use the Mappings page to map the input column names to the output column names.
- Drag and drop the available input columns to the available destination columns. The Mappings page can also be used to select an input column to ignore by setting the input column to <ignore>.



The Error Output Page

- Error
- you
- table
- com
- Not
- flow

The Error Output Page

OLE DB Source Editor

Configure the properties used by a data flow to obtain data from any OLE DB provider.

Connection Manager

Columns

Error Output

Input or Output	Column	Error	Truncation	Description
OLE DB Source O...				
	ProductID	Fail component	Fail component	Conversion
	ProductName	Fail component	Fail component	Conversion
	ProductColor	Fail component	Fail component	Conversion
	ProductSize	Fail component	Fail component	Conversion
	ProductSellingStartDate	Fail component	Fail component	Conversion
	ProductSellingEndDate	Fail component	Fail component	Conversion
	ProductSellingEndDateText	Fail component	Fail component	Conversion
	ProductCategoryID	Fail component	Fail component	Conversion
	ProductCategoryName	Fail component	Fail component	Conversion
	ParentProductCategoryName	Fail component	Fail component	Conversion

Error Flows

- The Error Output component, the Configure Error path to redirect

The diagram illustrates an error flow in a data integration tool. It shows an 'OLE DB Source' component connected to an 'OLE DB Destination' component. A dashed line indicates an error path from the source to an 'ERROR OLE DB Destination' component, which is marked with a red 'X'.

The 'Configure Error Output' dialog box is open, showing the configuration for the 'OLE DB Source' component. The dialog box has a title bar with standard window controls. Below the title bar, there is a description: 'Specify how row-level errors are handled by this component. You can handle errors in the row, or truncation errors in columns. Errors can fail the component, or they can be ignored, or they can be redirected to an error output.'

The main area of the dialog box contains a table with the following columns: 'Input or Output', 'Column', 'Error', 'Truncation', and 'Description'. The table lists several columns from the 'OLE DB Source' component, each with a corresponding error and truncation handling configuration.

Input or Output	Column	Error	Truncation	Description
OLE DB Sour...	ProductID	Fail component	Fail component	Conversion
	ProductName	Fail component	Fail component	Conversion
	ProductColor	Fail component	Fail component	Conversion
	ProductSize	Fail component	Fail component	Conversion
	ProductSelling...	Fail component	Fail component	Conversion
	ProductSelling...	Fail component	Fail component	Conversion
	ProductSelling...	Fail component	Fail component	Conversion
	ProductCateg...	Fail component	Fail component	Conversion
	ProductCateg...	Fail component	Fail component	Conversion
	ParentProduct...	Fail component	Fail component	Conversion

Below the table, there is a section labeled 'Set this value to selected cells:' with a dropdown menu set to 'Fail component' and an 'Apply' button. At the bottom right of the dialog box, there are 'OK' and 'Cancel' buttons.

DEMO 4.
DATA FLOWS OVERVIEW.
DATA FLOW SOURCE.
DATA FLOW PATH

Data Flow Transformations

Transformations are the third and final component to consider when working with data flows. The following are types of data flow transformations:

- **The Sort transformation:** performs single or multiple (numbered) sorts on input data,
- **The Data Conversion transformation:** converts input column data to a different data type and inserts it into a new output column.
- **The Aggregate transformation:** performs aggregate functions and calculations to column values (or values that have been grouped using a GROUP BY clause, and copies the results to the output.
- **The Derived Column transformation:** creates new column values or replaces existing values by applying expressions that can contain any combination of variables, functions, operators and columns.
- **The Lookup transformation:** performs lookups by joining data in input columns with columns in a reference dataset. You use the lookup to access additional information in a related table that is based on values in common columns.
- **The Union All transformation and The Merge transformation:** combines multiple inputs into a single output. The Merge transformation (included for backward compatibility) acts like the Union All transformation, but is limited to two inputs, and it requires those inputs to be sorted.
- **The Merge Join transformation:** joins two sorted datasets using a FULL, LEFT, or INNER join before copying to the output.
- **NOTE:** When possible, it is recommend performing these transformations in the Data Flow's data sources .

DEMO 5.
Sort, Data conversion,
Derived Column.

Tuning Data Sources

SSIS is a powerful tool that can perform many different tasks, but that flexibility comes at a cost in performance. You can create more efficient code to manipulate data in files or database, using languages like Python or SQL, at the cost of losing the visual workflow of your ETL process.

- **Avoid pulling all the data** from the source if you only need a part of it. This makes a big difference when working with tables or files containing sequential data. For example, a web server's log file would have new entries each day, but existing entries might not ever be updated. Therefore, it will increase performance by restricting data to only be loaded from updated columns.
- **Use Sequence containers** to process data in parallel, which will help to reduce overall ETL execution time at the cost of the computers resources (RAM and CPU).
- **Avoid transforming large amounts of data directly from a file.** Often it is faster to import data into a temporary (staging) table and then use SQL transformation code to complete your ETL process.
- **Avoid using SSIS Events to track progress.** Each event handler is a performance drain on the ETL execution. Instead consider using logging tables in combination with ETL stored procedures.
- **Consider removing indexes on the destination tables before loading it.** If the source data is not sorted before it is inserted, it may be best to drop indexes on a table before loading its data, and re-create the indexes after loading completes. Then let the database engine shuffles the data into its correct location as needed.
- **Avoid implicit conversion.** Instead, convert data outside of SSIS's own expression language runtime environment. For example, use the SQL language for data in a database, or use C# or Python for data in a file.

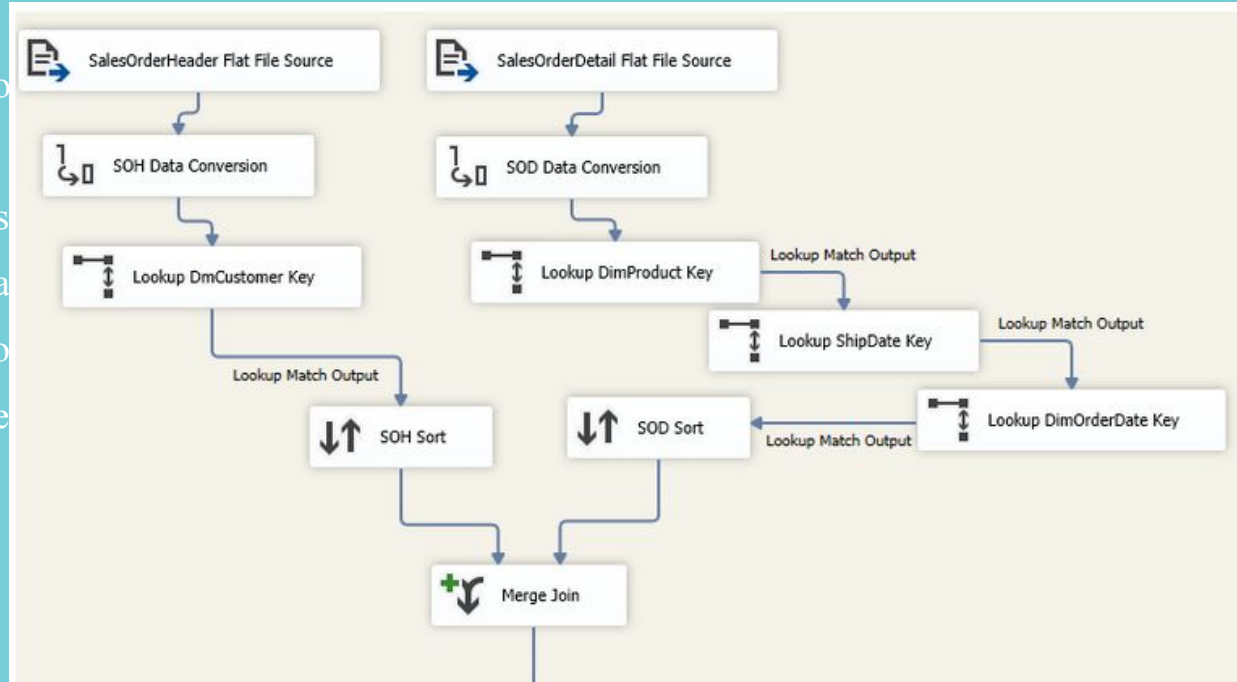
Staging Tables

One common part of ETL pro

This can be done in two ways

- The first is to import the data
- the second is to import it to

Although the first sounds like an initial setup cost.



4. DEPLOYMENT AND TROUBLESHOOTING

Troubleshooting Errors

Good ETL creation includes error handling and troubleshooting.

Microsoft includes a number of features in both SSIS and SQL Server that can help you troubleshoot ETL processing:

- using SSIS Error Paths and Event handlers,
- setting up ETL logging,
- different ways to deploy SSIS packages,
- automate ETL processing using SQL Server Agent.

Handling Data Flow Errors with Error Paths

Error paths are represented as red connecting lines between data flow components. Rows of

data that have

all data flow

without a p

Configuring

- Fail Com

- Ignore Fa

its output

- Redirect Ro

Data Flow and Error Paths

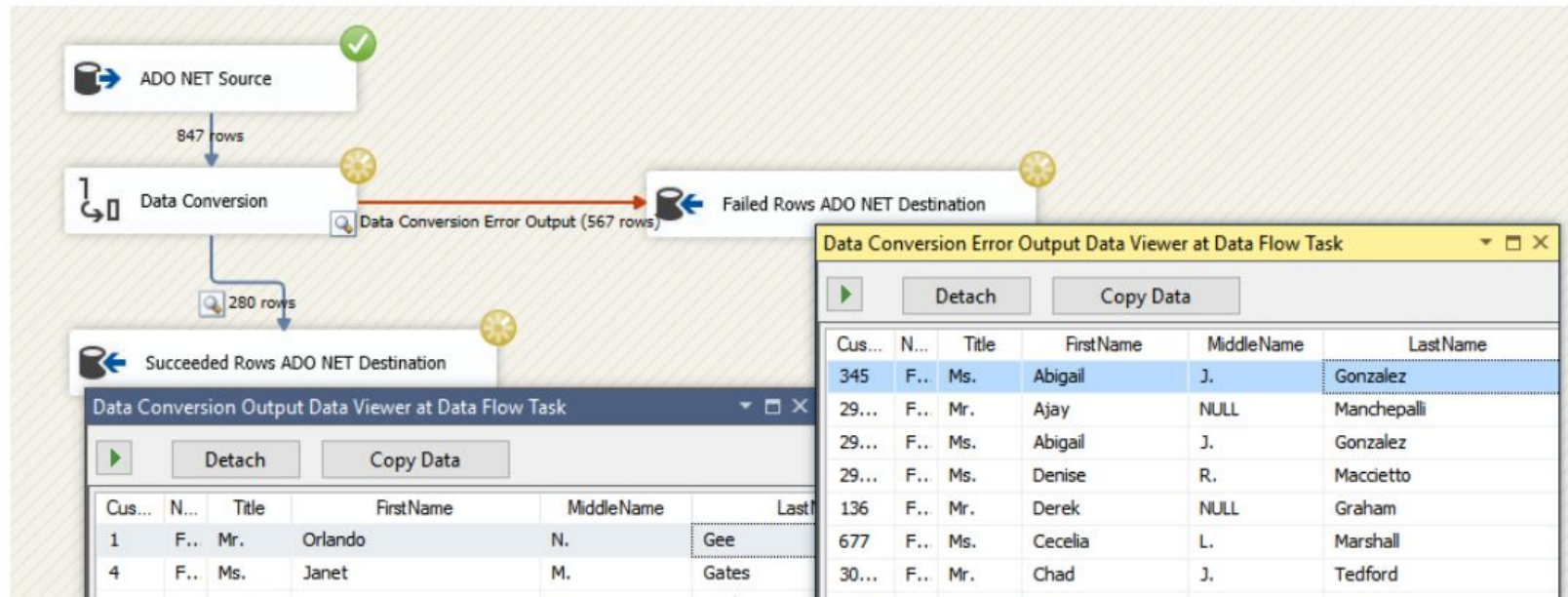
Configure Error Output

Specify how row-level errors are handled by this component. You can handle errors in the row, or truncation errors in columns. Errors can fail the component, or they can be ignored, or they can be redirected to an error output.

Input or Output	Column	Error	Truncation	Description
Data Conversion Output	Copy of LastName	Redirect row	Redirect row	Conversion

Troubleshooting Data Flow Issues with Data Viewers

Using Data Viewers



Event Handlers

Adding an Event Handler

- Control Flow
- Error Handling
- Data Flow
- Parameters
- Event Handlers
- Package Explorer

They include
a package

Executable:

Adding a Script Task to handle Control Flow errors

DimProducts Data



ErrorPaths.dtsx [Design]*

Control Flow Data Flow Parameters Event Handl...

Package Ex...

Execution R...

Executable:

Event handler:

DimCustomers Data Flow Task

OnError

Delete

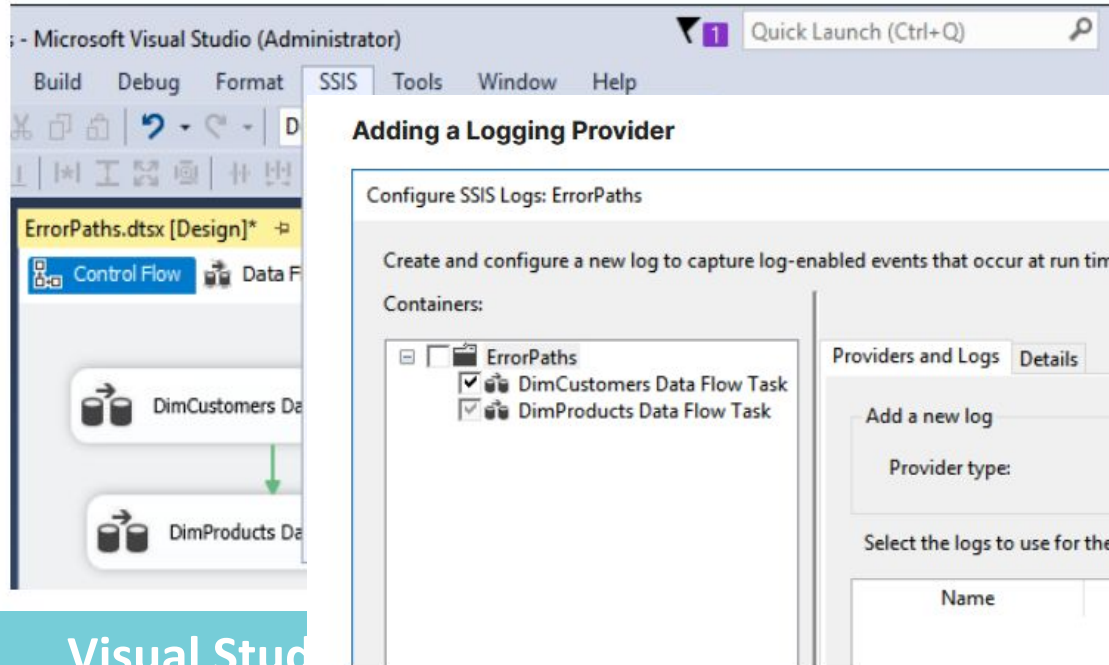


Log OnError Data for DimCustomers Script Task

- To add an event handler to a task, follow these steps:
1. Select the task in the Package Explorer.
 2. Right-click the task and select 'Add Event Handler'.
 3. Select the event handler to add from the list.
 4. Configure the event handler properties.

Logging SSIS Packages

Adding logging to a SSIS package



events on packages,

Visual Studio

Configuring a Log Provider

EPUALVIW01F9 (SQL Server 15.0.2000.5 - KYIV\Andrii_Pavli)

select * from [dbo].[sysssislog]

110 %

id	event	computer	operator	source	sourceid	executionid	starttime	endtime
1	PackageStart	EPUALVIW01F9	KYIV\Andrii_Pavlish	Demo6 TroubleShooting	DCDE186C-2679-4F31-9951-2FAD0BAC810C	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
2	OnPreExecute	EPUALVIW01F9	KYIV\Andrii_Pavlish	Demo6 TroubleShooting	DCDE186C-2679-4F31-9951-2FAD0BAC810C	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
3	OnPreExecute	EPUALVIW01F9	KYIV\Andrii_Pavlish	Data Flow Task - DimProductCategory	CB73B3C5-46F9-4B84-B37C-D3434367D9AE	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
4	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Data Flow Task - DimProductCategory	CB73B3C5-46F9-4B84-B37C-D3434367D9AE	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
5	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Demo6 TroubleShooting	DCDE186C-2679-4F31-9951-2FAD0BAC810C	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
6	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Data Flow Task - DimProductCategory	CB73B3C5-46F9-4B84-B37C-D3434367D9AE	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
7	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Demo6 TroubleShooting	DCDE186C-2679-4F31-9951-2FAD0BAC810C	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
8	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Data Flow Task - DimProductCategory	CB73B3C5-46F9-4B84-B37C-D3434367D9AE	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
9	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Demo6 TroubleShooting	DCDE186C-2679-4F31-9951-2FAD0BAC810C	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
10	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Data Flow Task - DimProductCategory	CB73B3C5-46F9-4B84-B37C-D3434367D9AE	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
11	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Demo6 TroubleShooting	DCDE186C-2679-4F31-9951-2FAD0BAC810C	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
12	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Data Flow Task - DimProductCategory	CB73B3C5-46F9-4B84-B37C-D3434367D9AE	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
13	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Demo6 TroubleShooting	DCDE186C-2679-4F31-9951-2FAD0BAC810C	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000
14	OnInformation	EPUALVIW01F9	KYIV\Andrii_Pavlish	Data Flow Task - DimProductCategory	CB73B3C5-46F9-4B84-B37C-D3434367D9AE	6716AD56-65F2-4B52-AC48-8D0EF0DC93C3	2021-01-19 21:46:16.000	2021-01-19 21:46:16.000

view, then the log
be logged.

that are to

File Connection Manager Editor

Configure the file connection properties to reference a file or a folder that exists or is created at run time.

Usage type: Create file

File: C:\ETLwithSSIS\Module05\LoggingDemo.csv

Browse...

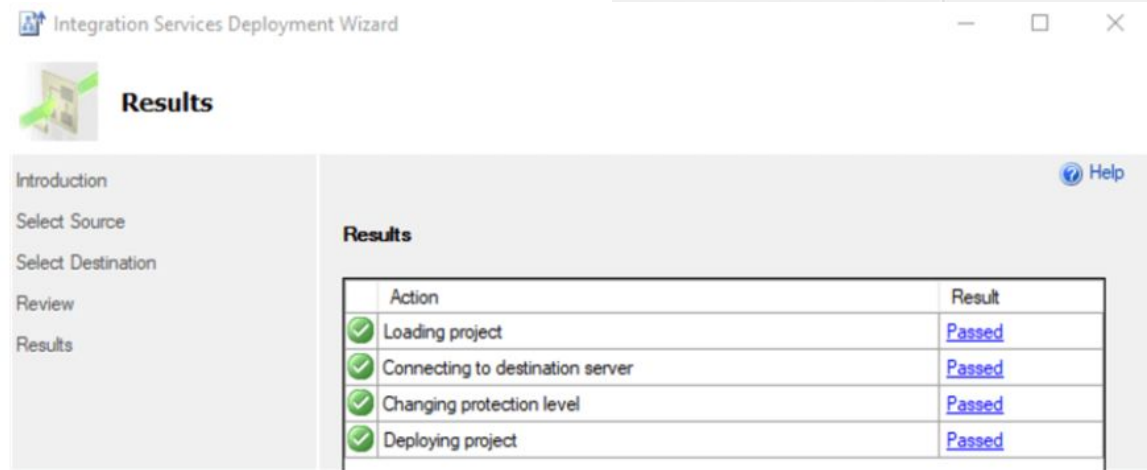
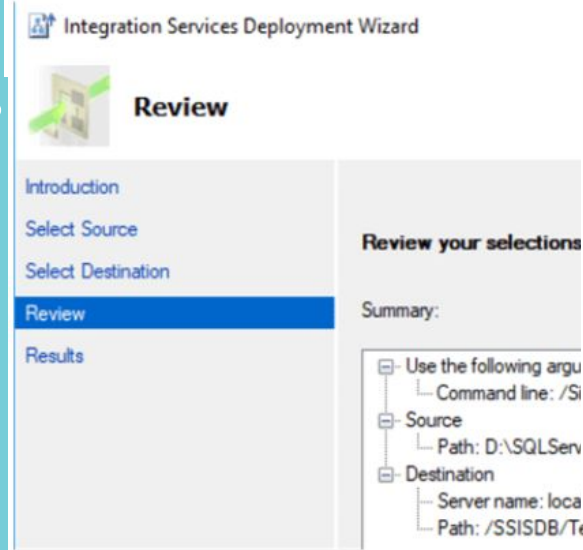
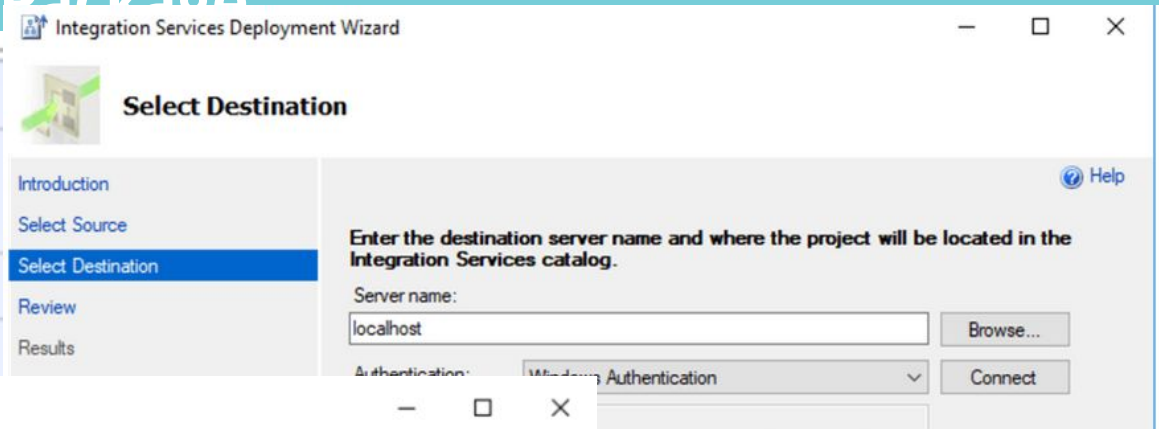
OK Cancel

Delete

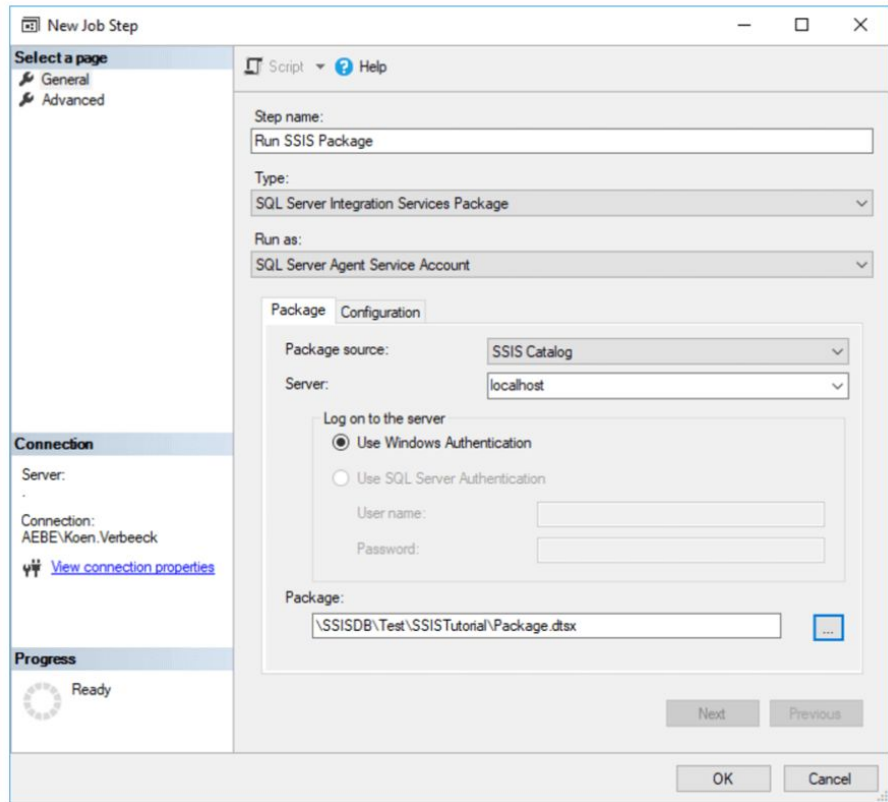
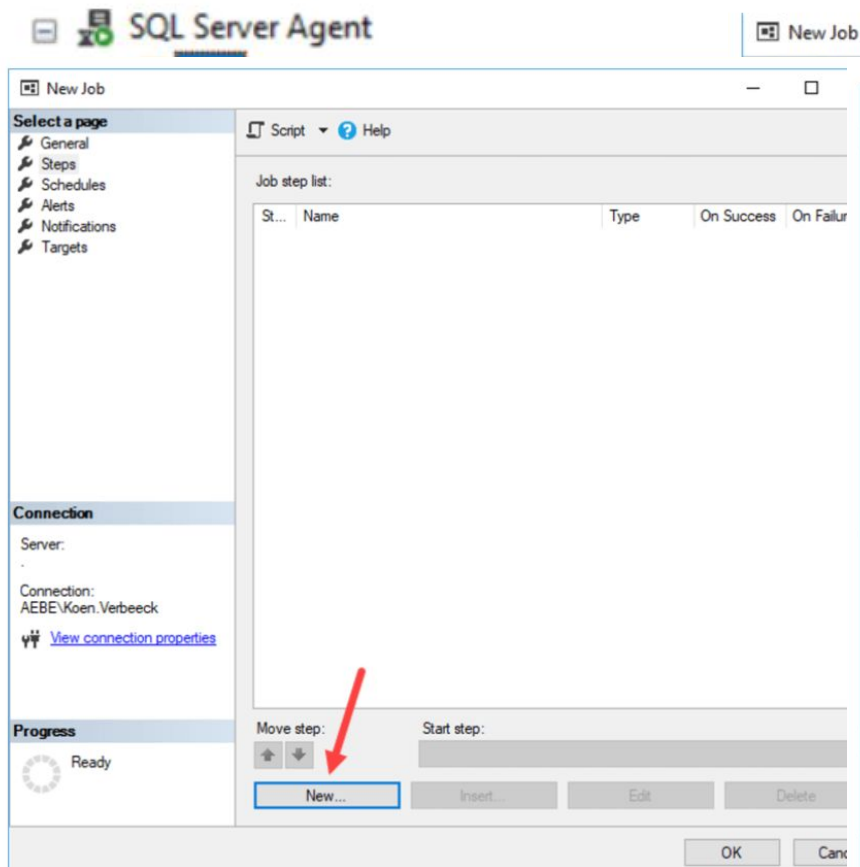
To configure unique logging options for this container, enable logging for it in the tree view.

OK Cancel Help

Deploying the SSIS Package



ETL Automation using SSIS Jobs



DEMO 6.
Troubleshooting and error handling.

Homework

THANK YOU!