Module 04 Notes

Data Movement – Part 2 (Visual ETL Processing)

# ****Module Overview****

In this module, you learn about how **data is moved between relational databases using a Visual ETL tool like SQL Server Integration Services (SSIS)**. This process involves **extracting source data, transforming** that data, and loading data into its destination, but this time we are adding a visual aspect. We will use an **example of a data warehouse**, but the **techniques shown apply to many scenarios**.

# ****Required Software****

Install SQL Developer Edition (or equivalent), Visual Studio 2019 Community edition with the SSIS Extension.

# Assignment

Each week you have to perform an assignment. Let us review this week's assignment.

# SSIS

"SSIS can **simplify your ETL process** by providing the following:

* A **visual** overview of the ETL process
* Visual and programmatic **control of a sequence** of ETL tasks
* A way to collectively schedule **automations**
* The ability to **individually execute tasks and track processing**
* **Access to any data source** that supports OLE DB or ODBC connections

SSIS is a platform that can perform **various tasks pertaining to data collection and automating processes**. It features tools that can be extremely helpful in performing the ETL process. Although the ETL process is its main function, it has several other features as well, such as the **ability to run .NET code, interact with the file system, and access various operating system resources**."1

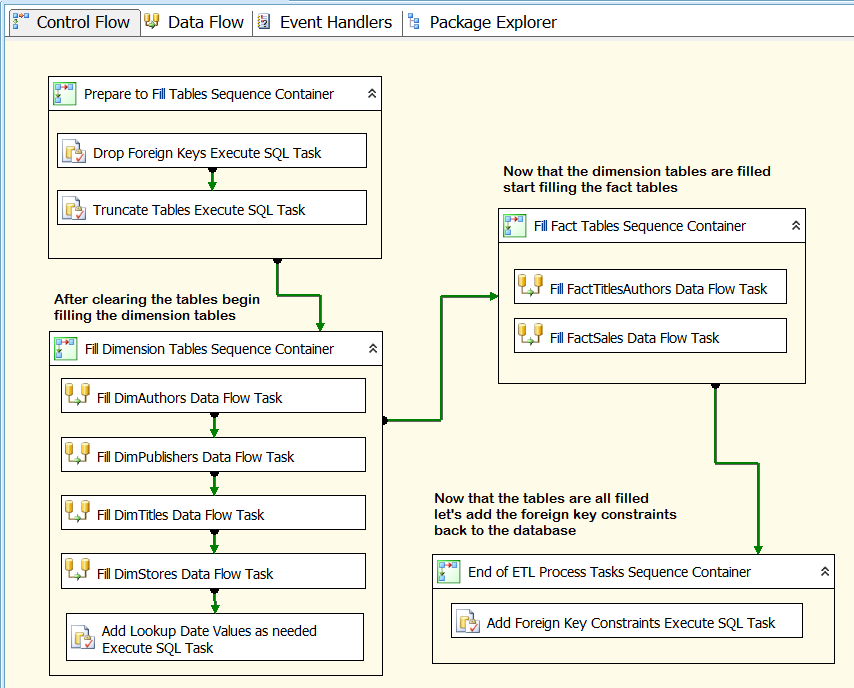


Figure: An ETL SSIS package

"Integration Services includes a rich **set** **of built-in tasks and transformations**; **tools for constructing packages**; and the Integration Services service for running and managing packages. You **can use** the graphical Integration Services tools to create solutions **without writing** a single line of **code**; **or** you can program the **extensive** Integration Services object model to create packages programmatically and **code custom tasks** and other package objects." (<https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services>, 2018)

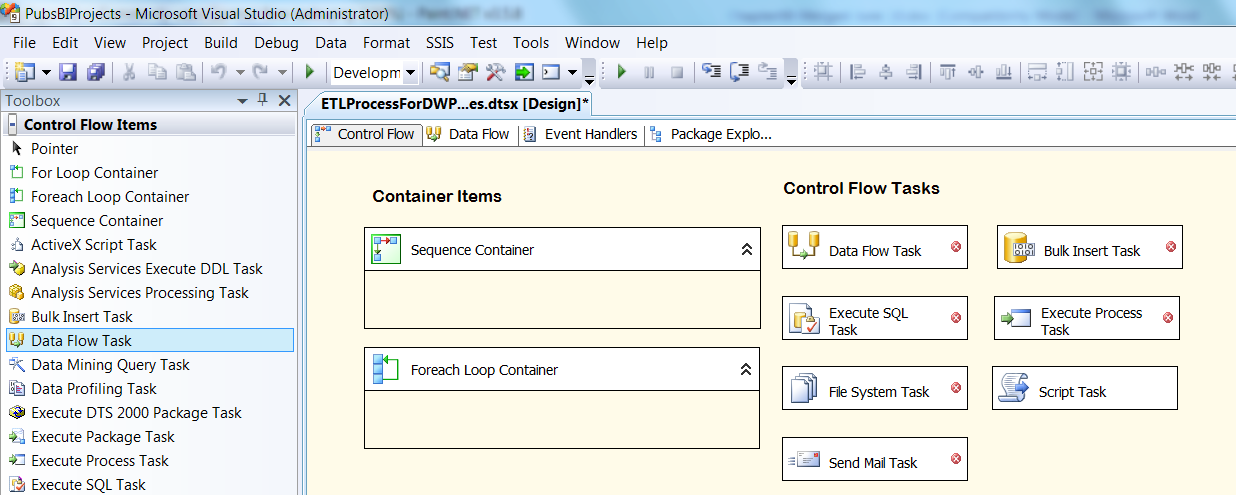


Figure: Commonly used Task in SSIS

# SSIS Projects

**SSIS projects or a collection of one or more code files** that perform various tasks. These tasks can widely range from **simple file imports and exports to complex management and automation tasks**. The code files are programmed in an XML language, and you could use any text editor to create them.

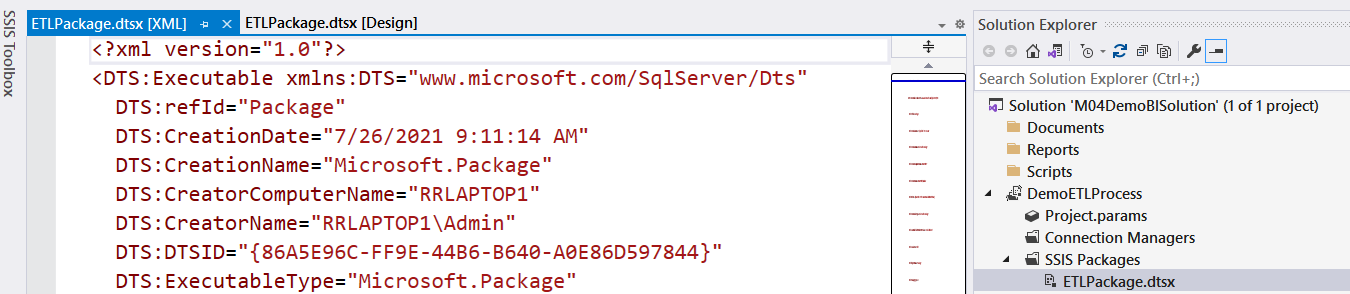


Figure the code in an SSIS package file

However, it would be unlikely for you to do so since Visual Studio's SSIS project will allow you to **create these code files** **by dragging and dropping objects from a toolbox** onto the surface of the code files. The **code files are known as Package files** and are represented as a visual surface. The package file has a "dtsx" extension, which came from SSIS predecessor called Data Transformation Services.

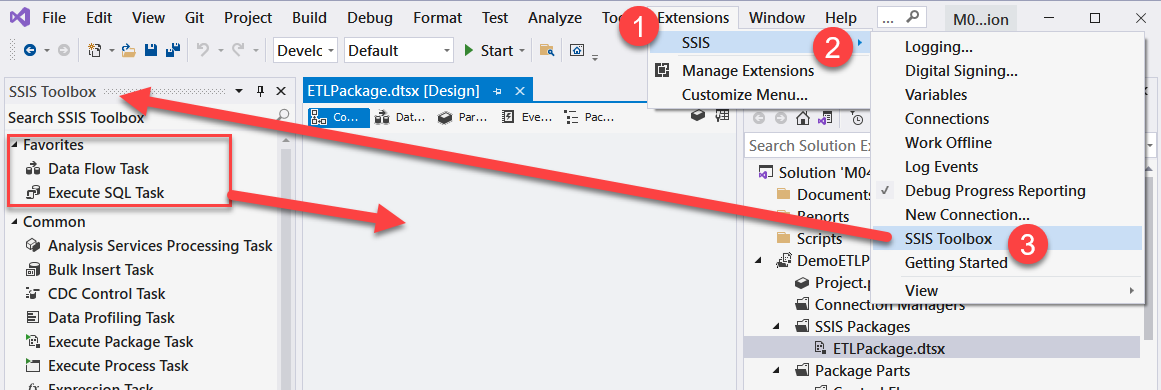


figure: the design surface of an SSIS package file

There are **several different tasks on the SSIS toolbox and a few containers**. I recommended all students **start with or the sequence containers** to organize their SSIS packages.

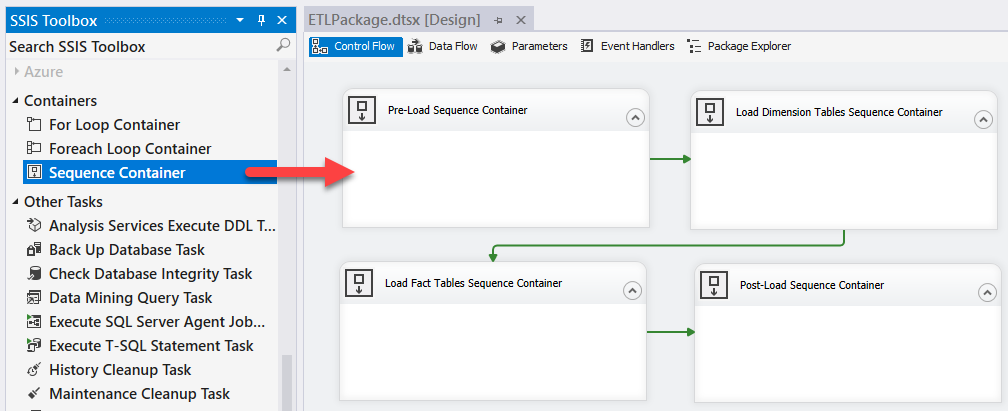


Figure: using Sequence Containers to organize your SSIS package file

## Demonstration – Creating an SSIS Project

In this demo, you review a SQL script that creates a source database and fills it with data generated on the internet and from an existing relational database.

**Files needed:** None

# SSIS Tasks add containers

The SSIS Toolbox included many tasks, but the most used ones are **Execute SQL Task and the Data Flow Task**. Let's look at each.

## Execute SQL Tasks

*"An* ***Execute SQL task allows you to run SQL code*** *on a connected database. For example, to use an Execute SQL task designed to drop foreign key constraints, you configure it to connect to the proper database by selecting the appropriate connection manager and then add SQL code to drop the constraints"1*

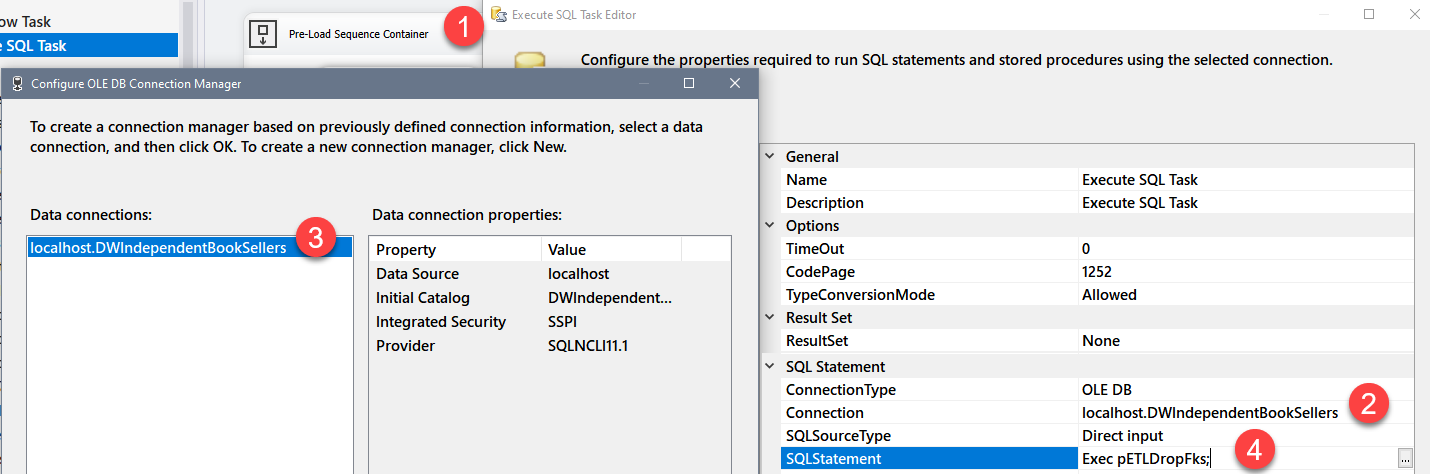


Figure: Configuring an Execute SQL task in SSIS

*"The* ***connections*** *available in the dropdown box are* ***context sensitive to the connection type*** *selected in the ConnectionType property. If the connection type is not set to OLE DB but instead is set to another connection type such as ADO.NET, any OLE DB connection managers that have been created will not appear in the dropdown selection. Switching the ConnectionType property to OLE DB allows them to appear.*

*To add SQL code, you need* ***to configure the SQL Statement property****. After clicking this property, you will see that an ellipsis button appears. When you* ***click the ellipsis button, a dialog appears where you can add your SQL code****. Hidden buttons like this one are becoming commonplace in Microsoft applications. If you do not immediately see a way to configure a property, try clicking it to see whether it contains a hidden button.*

*Tip:* ***In Microsoft’s most recent user interfaces, dropdown boxes, buttons, or ellipsis icons do not appear until you click a property setting.****"1*

## Demonstration – Using Execute SQL Tasks

In this demo, you review SQL scripts that create a source database, a destination database, and ETL objects. Will then use Execute SQL tasks to run the ETL sprocs.

**Files needed:**

* 0\_InstPubs.sql
* 1\_SourceDatabase.sql
* 3\_DWDatabase.sql
* 5\_ETLObjects.sql

## Data Flow Tasks

"**Data Flows are the most complex of the SSIS tasks because they are made up of a composite of many subcomponents**. There are three different types of data flow subcomponents: **sources, transformations, and destinations.** Most often sources pull data from tables and views, but you can also extract data from multiple file types, and even SSAS cubes. Transformations modify, summarize, and clean data. Destinations load data into tables, cubes, files, or in-memory datasets that can be used by other tasks within the SSIS package itself. **Data flows are configured using the Data Flow tab.** You can access this tab at the top of the package designer window or right-click any data flow task and select Edit from the context menu. Either way, the user interface navigates to the Data Flow tab."1

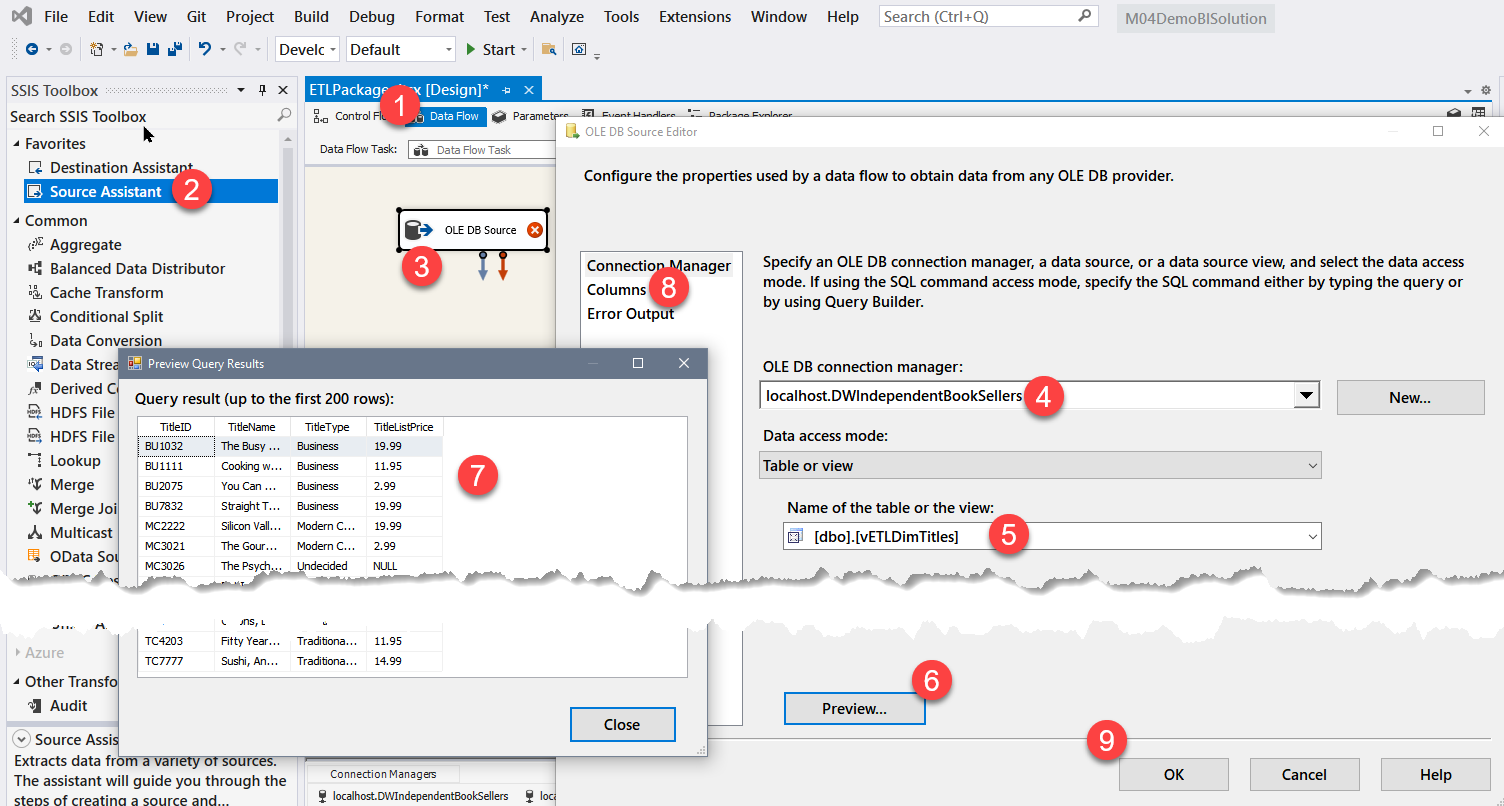


Figure: Configuring a Data Flow task in SSIS

## Executing the Entire Package

"When you have completed configuring and testing all of the tasks in your SSIS package, you need to **test the entire package as a whole. This vital step is often missed by developers**, and they only find out later that the package does not work as a unit—usually while giving a presentation showing how well it works!

The best way to **reset the database** back to its normal, preload state, is to use the SQL code script and then right-click the package in Solution Explorer to access the context menu."1

## Demonstration – Using Data Flow Tasks

In this demo, you review a SQL script that creates a source database and fills it with data generated on the internet and from an existing relational database.

**Files needed:** 0\_InstPubs.sql and 1\_SourceDatabase.sql

# Command Based ETL

One of the fastest ways to import external data is by using command-based ETL tools such as the following:

**BCP.exe (External application)**

A command console application that exports and imports data.

**BULK INSERT statement**

A Transact-SQL statement that imports data from a file

**INSERT ... SELECT \* FROM OPENROWSET(BULK...) statement**

A Transact-SQL statement that uses import data into a SQL Server table by using the OPENROWSET function to select external data

## Demonstration – Using Command Based ETL tools

In this demo, you review three common ways to import file data using command-based ETL tools.

**Files needed:** 6\_Command Based ETL Demos.sql

# References

1. Pro SQL Server 2012 BI Solutions, Randal Root and Cari Mason, Apress, 2012