



SQL Server SSIS Training

Customer Success Unit
Karen Ku (chiku@microsoft.com)



Agenda

day1

- Module 1: Introduction & Architecture
- Module 2: Development Tools
- Module 3: SSIS Development
- Module 4: Error Handling
- Hands-On Lab

day2

- Module 5: Deployment
- Module 6: Troubleshooting
- Module 7: Best Practices
- Hands-On Lab

Demo 環境

- Target SQL Server : SQL Server 2016
- SSDT : Visual Studio 2019
- Sample SQL Database : <https://github.com/Microsoft/sql-server-samples/releases/download/adventureworks/AdventureWorksDW2012.bak>
- 範例資料:
- <https://github.com/karenchiku/Creating-a-Simple-ETL-Package>



Module 1: Introduction & Architecture

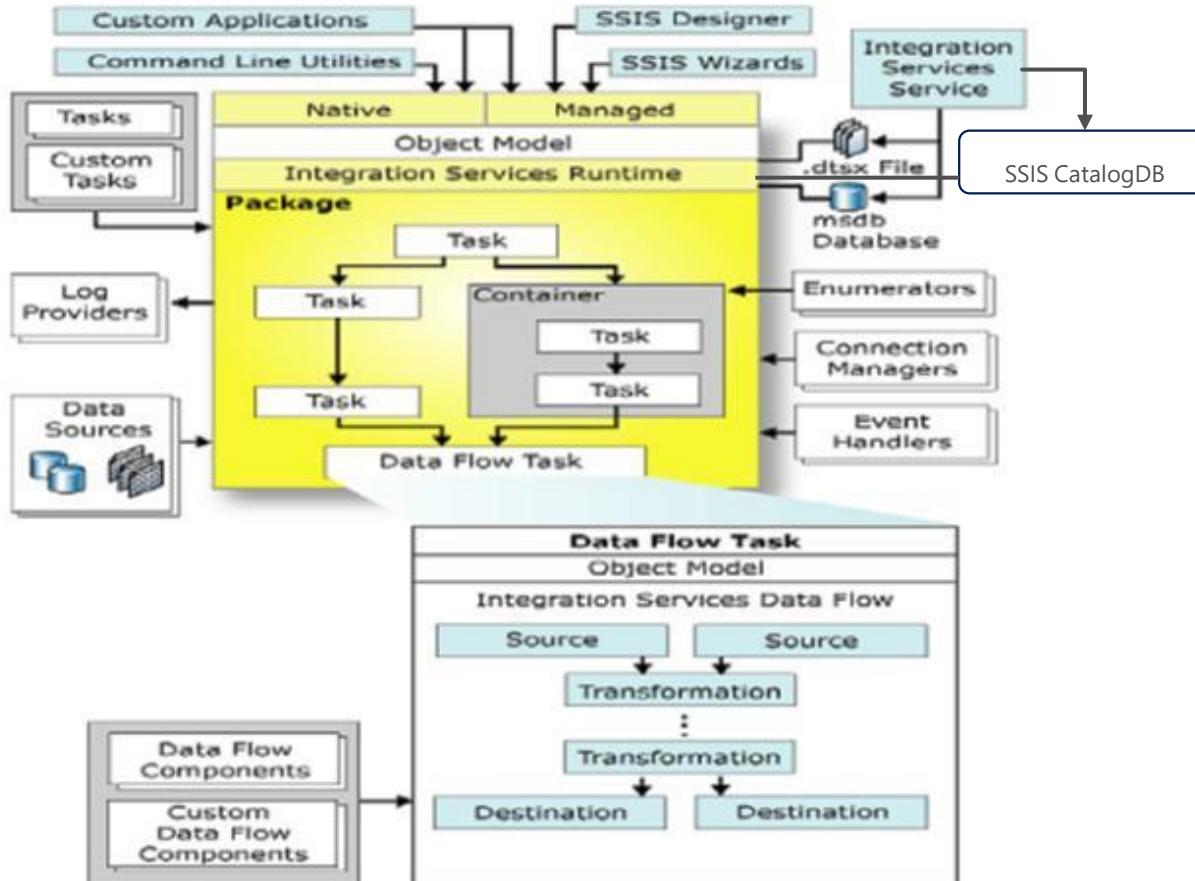
Agenda

- What is SQL Server Integration Services (SSIS)?
- SSIS Architecture
- SSIS Components
- SSIS High Availability
- New to SSIS 2016, SSIS 2017, and SSIS 2019.
- SSIS Usage Scenario

What is SQL Server Integration Services (SSIS)?

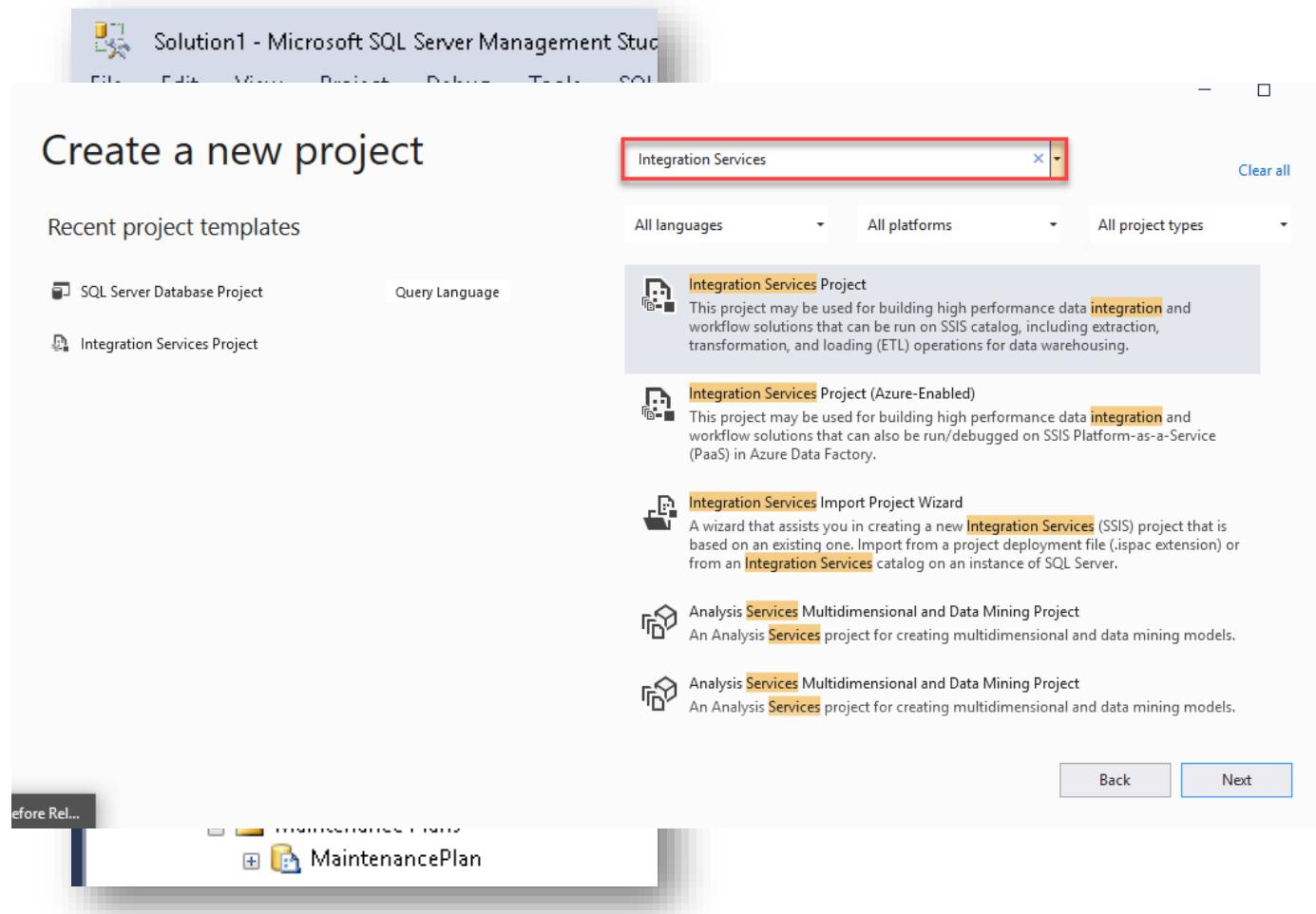
- SQL Server Integration Services (SSIS) provides a platform for Extract, Transform, and Load (ETL) functionality for Enterprise data.
- Introduced in SQL Server 2005.
- Successor of Data Transformation Services (DTS) (SQL Server 2000 or older).

SSIS Architecture



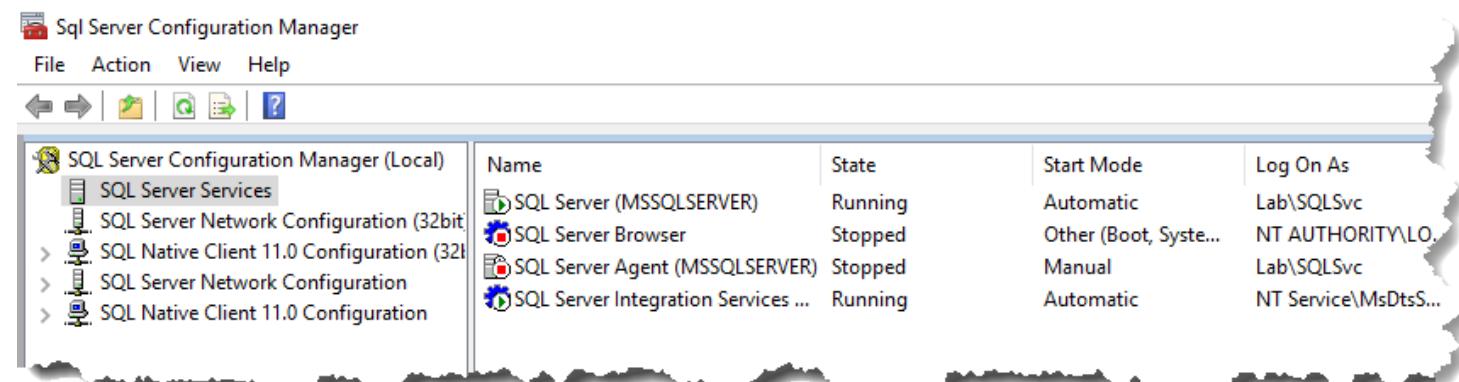
SSIS Components

- Development & Management Tools
 - SQL Server Management Studio (SSMS)
 - SQL Server Data Tools (SSDT) Add-On for Visual Studio



SSIS Components

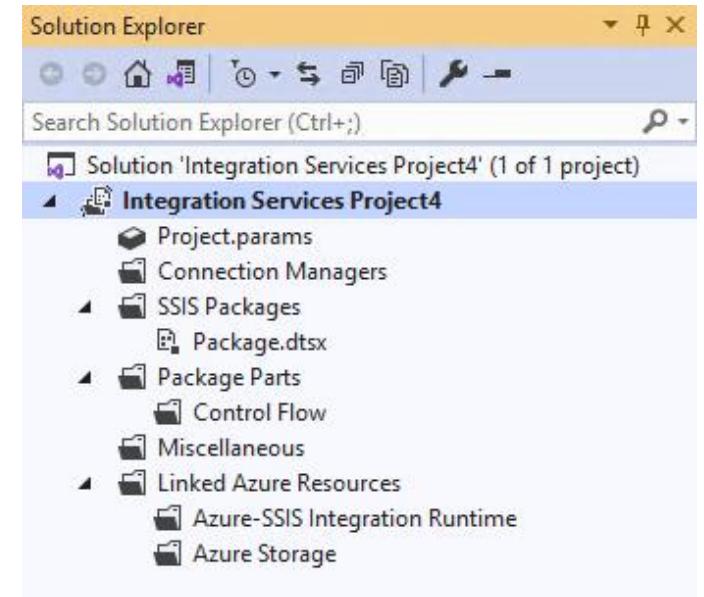
- SSIS Service
 - Extends the management capabilities of SSMS.
 - Generally installed on SQL Server instance.
 - Visible in SQL Server Configuration manager or SSMS.
 - Required for SQL Server SSIS DB Catalog.
 - Import/Export SSIS Packages.
 - Manually Execute/Stop SSIS Packages.
 - Provides ability to monitor executions.



SSIS Components

- Project & Packages

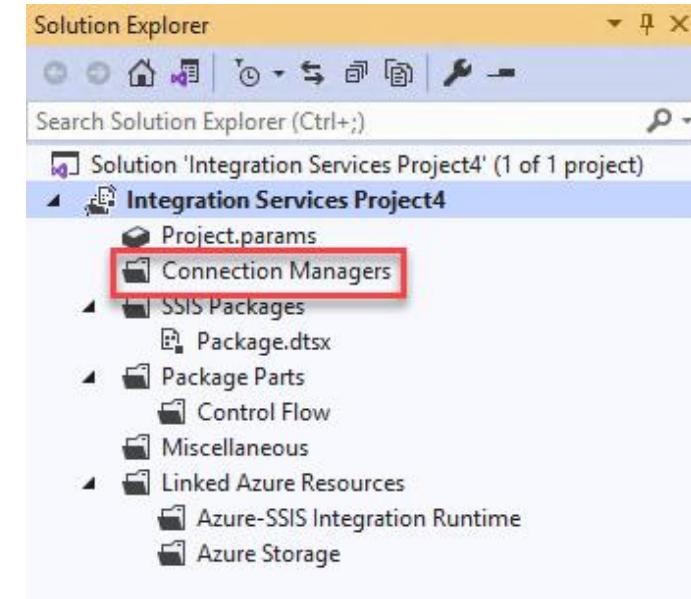
- Projects allow the ability group packages and objects with similar workload together.
- Projects can contain packages, connection manager (shared connections), parameters, etc.



Tip: If solution explorer is not visible, you can launch via View Menu > Solution Explore or keyboard shortcut CTRL + W, S

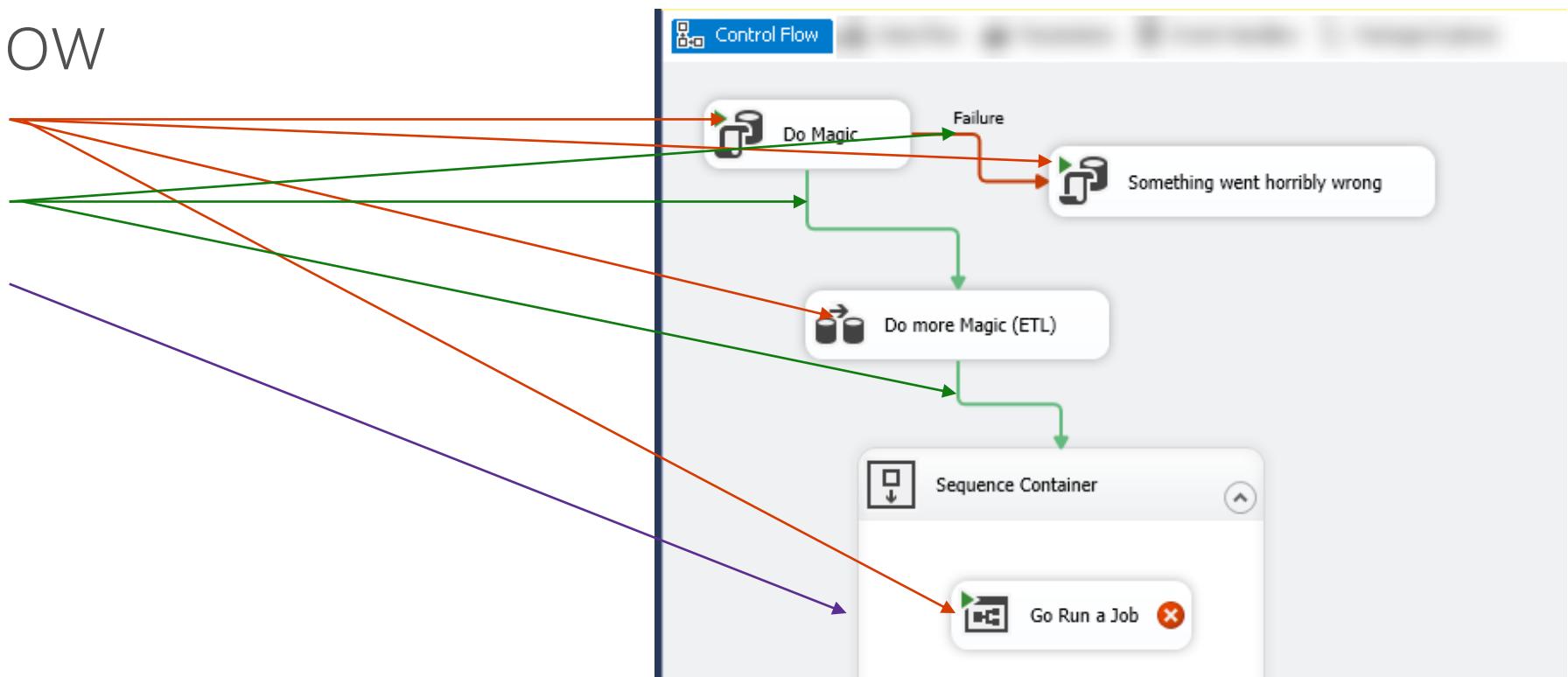
SSIS Components

- Connection Managers
 - Provide ability to collect to data sources for both source and destination.
 - Ability to create connections both at project level and package level.



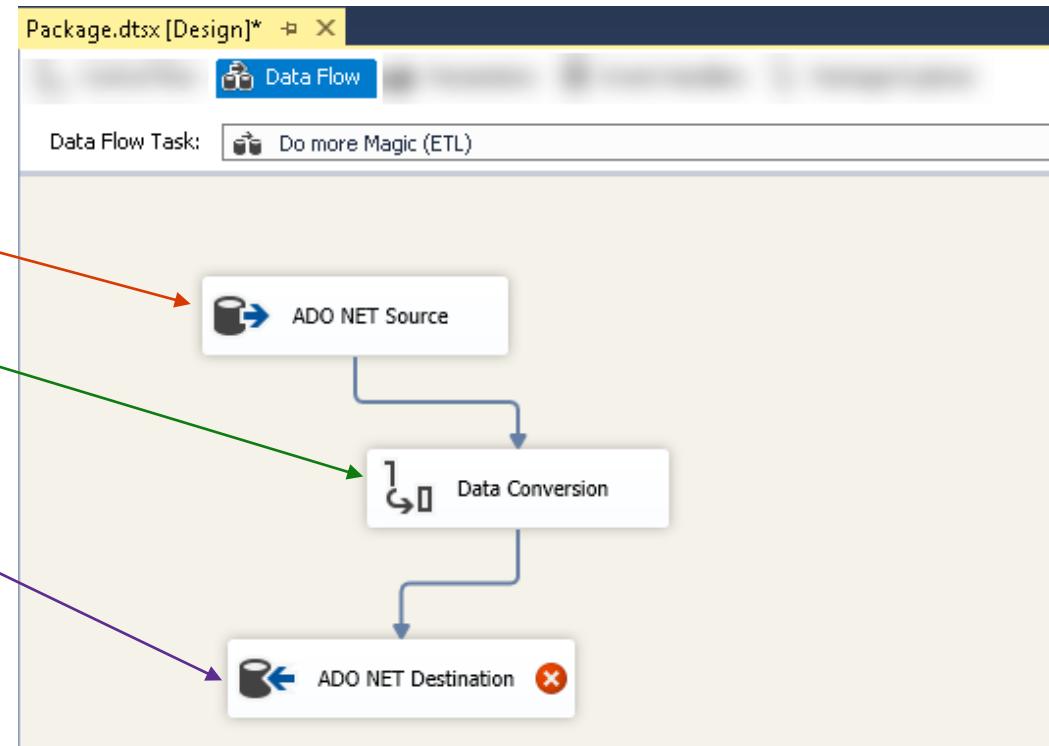
SSIS Components

- Control Flow
 - Tasks
 - Constraints
 - Containers



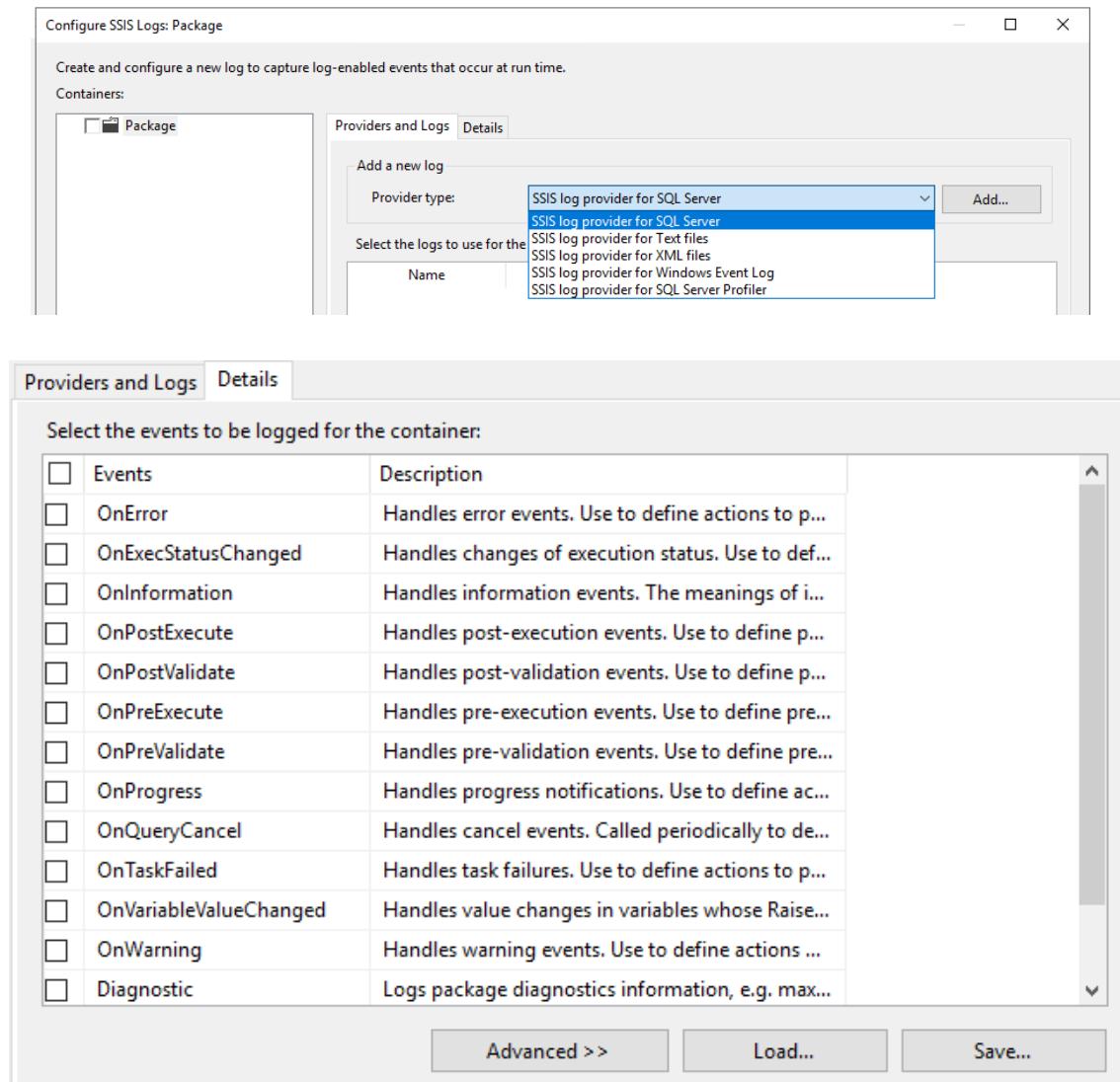
SSIS Components

- Data Flow
 - Extract
 - Transform
 - Load



SSIS Components

- Log Providers
 - Define destination for logging execution log details.
 - Information can be logged to:
 - Text File
 - SQL Profiler
 - SQL Server
 - Windows Event Logs
 - XML Files
 - We can choose which events & details to log.



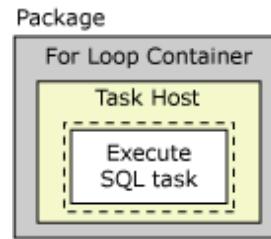
SSIS Components

- Enumerators
 - Allows you to loop through a list of objects.

Enumerator	Configuration requirements
Foreach ADO	Specify the ADO object source variable and the enumerator mode. The variable must be of Object data type .
Foreach ADO.NET Schema Rowset	Specify the connection to a database and the schema to enumerate.
Foreach File	Specify a folder and the files to enumerate, the format of the file name of the retrieved files, and whether to traverse subfolders.
Foreach From Variable	Specify the variable that contains the objects to enumerate.
Foreach Item	Define the items in the Foreach Item collection, including columns and column data types.
Foreach Nodelist	Specify the source of the XML document and configure the XPath operation.
Foreach SMO	Specify the connection to a database and the SMO objects to enumerate.
Foreach HDFS File Enumerator	Specify a folder and the files to enumerate, the format of the file name of the retrieved files, and whether to traverse subfolders.
Foreach Azure Blob	Specify the Azure blob container that contains blobs to be enumerated.
Foreach ADLS File	Specify the Azure Data Lake Store directory that contains the files to be enumerated.
Foreach Data Lake Storage Gen2 File	Specify the Azure Data Lake Storage Gen2 directory that contains the files to be enumerated, along with other options.

SSIS Components

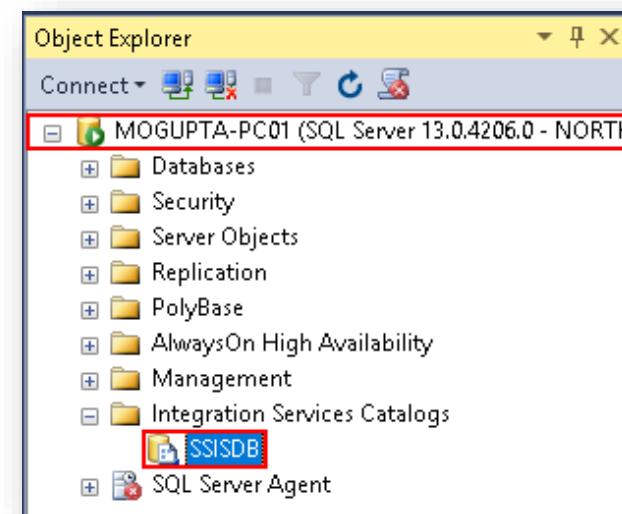
- Event Handlers
 - Triggers at various stages of package execution.
 - Can be used to ...
 - Clean up temporary data storages when package or task finishes running.
 - Retrieve system information to assess resource availability before a package runs.
 - Refresh data in a table when a lookup in a reference table fails.
 - Send an e-mail message when an error or a warning occurs or when a task fails.



Event handler	Event
OnError	The event handler for the OnError event. This event is raised by an executable when an error occurs.
OnExecStatusChanged	The event handler for the OnExecStatusChanged event. This event is raised by an executable when its execution status changes.
OnInformation	The event handler for the OnInformation event. This event is raised during the validation and execution of an executable to report information. This event conveys information only, no errors or warnings.
OnPostExecute	The event handler for the OnPostExecute event. This event is raised by an executable immediately after it has finished running.
OnPostValidate	The event handler for the OnPostValidate event. This event is raised by an executable when its validation is finished.
OnPreExecute	The event handler for the OnPreExecute event. This event is raised by an executable immediately before it runs.
OnPreValidate	The event handler for the OnPreValidate event. This event is raised by an executable when its validation starts.
OnProgress	The event handler for the OnProgress event. This event is raised by an executable when measurable progress is made by the executable.
OnQueryCancel	The event handler for the OnQueryCancel event. This event is raised by an executable to determine whether it should stop running.
OnTaskFailed	The event handler for the OnTaskFailed event. This event is raised by a task when it fails.
OnVariableValueChanged	The event handler for the OnVariableValueChanged event. This event is raised by an executable when the value of a variable changes. The event is raised by the executable on which the variable is defined. This event is not raised if you set the RaiseChangeEvent property for the variable to False. For more information, see Integration Services (SSIS) Variables .
OnWarning	The event handler for the OnWarning event. This event is raised by an executable when a warning occurs.

SSIS Components

- SSIS CatalogDB
 - Objects stored here are projects, packages, parameters, environments, and operational history.



SSIS Components

- SSIS Reports
 - Provide execution history of all the packages in SSIS Catalog DB.

The screenshot shows the Integration Services Dashboard. At the top, there's a navigation bar with 'Integration Services Catalogs' (containing 'SSISDB' with a red notification badge '1'), 'SQL Server', and 'Active Operations'. Below the navigation bar is the title 'Integration Services Dashboard' and the date 'on MOGUPTA-PC01 at 5/8/2018 10:57:52 AM'. A descriptive text states: 'This report provides information about operations that have run in the past 24 hours, including executions that are currently running.' The main area displays 'Execution Information (Past 24 Hours)' with counts: Failed (0), Running (0), Succeeded (1), and Others (0). Below this is 'Package Information (Past 24 Hours)' with a chart showing one successful package named 'Properties'. To the right, under 'Other Integration Services Reports', are links for 'All Executions', 'All Validations', and 'All Operations'. At the bottom, a table lists the package 'Demo01\Integration Services Project1\Package.dtsx' with status 'Succeeded', count '0/1', and timestamp '5/8/2018 10:57:06 AM'.

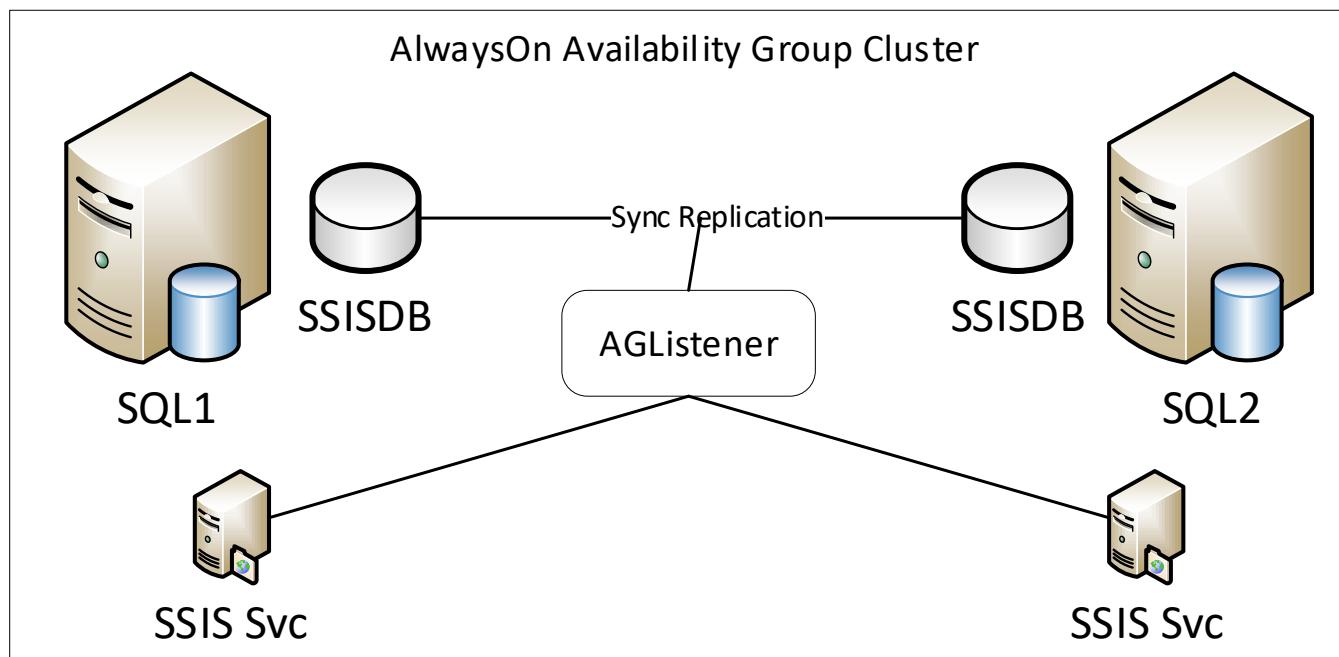
Succeeded	Overview	All Messages	Execution Performance	Demo01\Integration Services Project1\Package.dtsx	0/1	5/8/2018 10:57:06 AM
Green						

SSIS Catalog DB

Demo

SSIS High Availability

- Leverage SQL Server AlwaysOn Availability Groups
- SSIS Service itself is not highly available, instead we leverage SSIS CatalogDB via AG to provide HA.



New to SSIS 2016

- Manageability
 - AlwaysOn Support for SSIS Catalog
 - Incremental Package Deployment
 - Support for Always Encrypted for SSIS Catalog
- Debugging
 - New security roles.
 - New logging levels/server wide settings.
- Better Package Management
 - Project Upgrade
 - Reusable Control Flow Templates
- Connectivity
 - Expanded connector for Odata, Hadoop, SAP BW, Azure Cloud, etc.

New to SSIS 2017

- Scale Out
- Integration Services on Linux
- Connectivity Improvements

New to SSIS 2019

- New Features to Improve File Operations
 - Flexible File Task
 - Flexible File Source and Destination

SSIS Usage Scenario

- Archival of Data (Export)
- Loading of New Data (Import)
- Transferring data from one data source to another
- Data cleansing or Transformation of Dirty Data
- DBA Tasks like Purging old files or indexing a database

Questions?

- What are some of the components of SSIS?
SSIS Catalog DB, Control Flow, Data Flow, etc.
- Which phase of the ETL is usually the most expensive?
Transport is the most expensive phase, in most ETL processes.
- Can we create multiple SSIS Catalog DB per instance?
No limited to one SSIS catalog database per instance.
- What high availability option do we recommend for SSIS?
AlwaysOn Availability Groups



Module 2: Development & Management Tools

Agenda

- SQL Server Data Tools
- SQL Server Management Studio
- Command Line Tools for Management

SQL Server Data Tools (SSDT)

- Introduced in SQL Server 2012.
- Allows for development of projects related to SQL Server services. Such as Reporting Services, Analysis Services, and Integration Services.
- Decoupled from SQL Server binaries to allow for easy of management and upgradeability.
- Requires Visual Studio to run.

SSDT Interface

方案

專案

封裝

The screenshot displays the Microsoft SQL Server Data Tools (SSDT) interface. The main area shows the Control Flow designer for a package named 'Template.dtsx'. The control flow contains several tasks: 'Start Task', 'For loop', 'Script Task', 'Script Task', 'Script Task', and 'End Task'. The 'Solution Explorer' window on the right shows the project structure for 'SSIS DW -Template', including connection managers like 'DWDB.conmgr' and 'SSISDB.conmgr', and packages such as 'MASSB_HOSP_DB_ALL.dtsx' and 'MASSB_IC_CDHOST.dtsx'. The 'Toolbars' window on the far left lists various tools available in the interface.

方案

專案

封裝

方案總管

搜尋方案總管 (Ctrl+Shift+F)

解決方案 'SSIS DW -Template' (1 個專案, 共 1 個)

SSIS DW -Template

Project.params

連線管理員

- DWDB.conmgr
- DWDBDAS.conmgr
- MASS.conmgr
- ORA_AP.conmgr
- ORA_AP_OLE.conmgr
- ORA_HA.conmgr
- ORA_HD1.conmgr
- ORA_HM.conmgr
- ORA_HM_PROD.conmgr
- ORA_HU.conmgr
- SSISDB.conmgr
- STGDB.conmgr

SSIS 封裝

- MASSB_HOSP_DB_ALL.dtsx
- MASSB_IC_HOSP.dtsx
- MASSB_IC_CDHOST.dtsx
- MASSB_REAL_HOSP.dtsx
- Template 1.dtsx

方案總管 Git 變更

使用者入門 (SSIS)

載入使用者入門資訊時發生錯誤。遠端伺服器傳回一個錯誤: (503) 伺服器無法使用。

變數

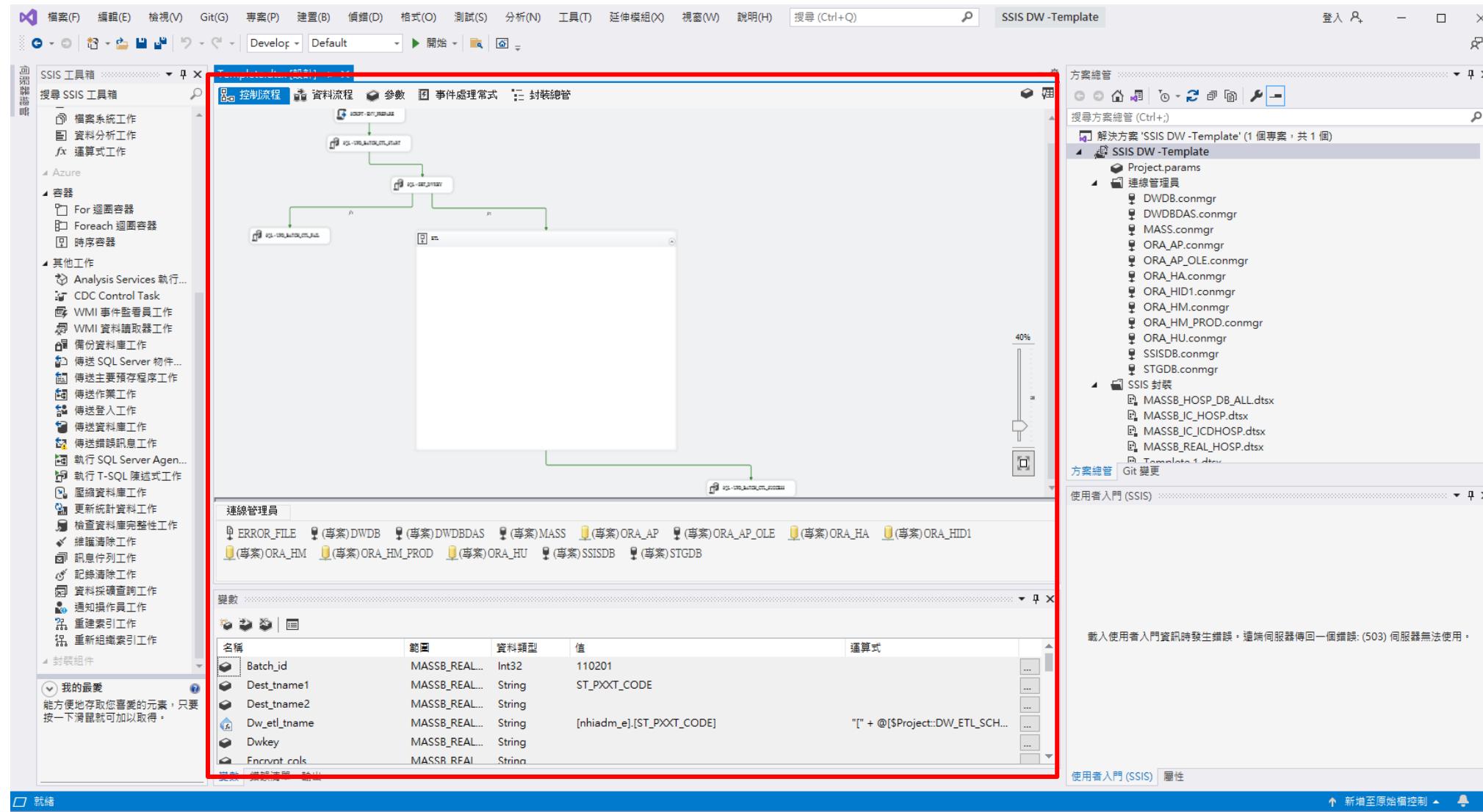
名稱	範圍	資料類型	值	運算式
Batch_id	MASSB_REAL...	Int32	110201	
Dest_tname1	MASSB_REAL...	String	ST_PXXT_CODE	
Dest_tname2	MASSB_REAL...	String		
Dw_etl_tname	MASSB_REAL...	String	[nihadm_e].[ST_PXXT_CODE]	"[" + @[\$Project:DW_ETL_SCH...
Dwkey	MASSB_REAL...	String		
Encrypt_cols	MASSB_REAL...	String		

變數 錯誤清單 輸出

使用者入門 (SSIS) 屬性

就緒 新增至原始檔控制

SSDT Interface



封裝

SSDT Interface

SSIS 工具箱

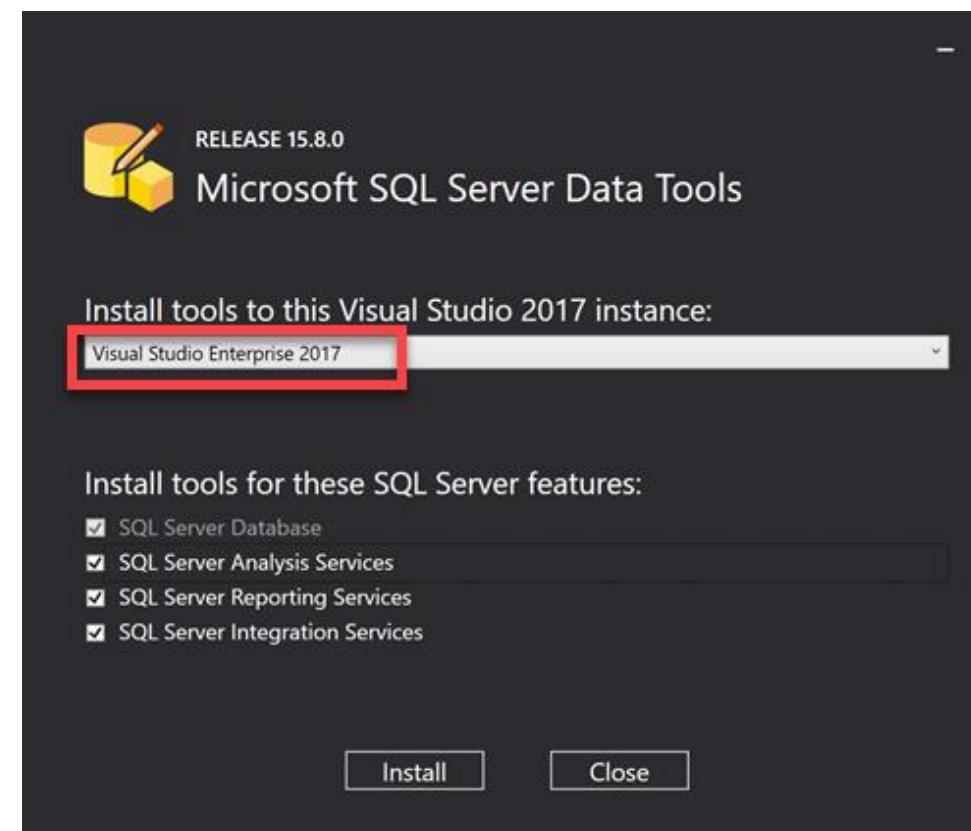
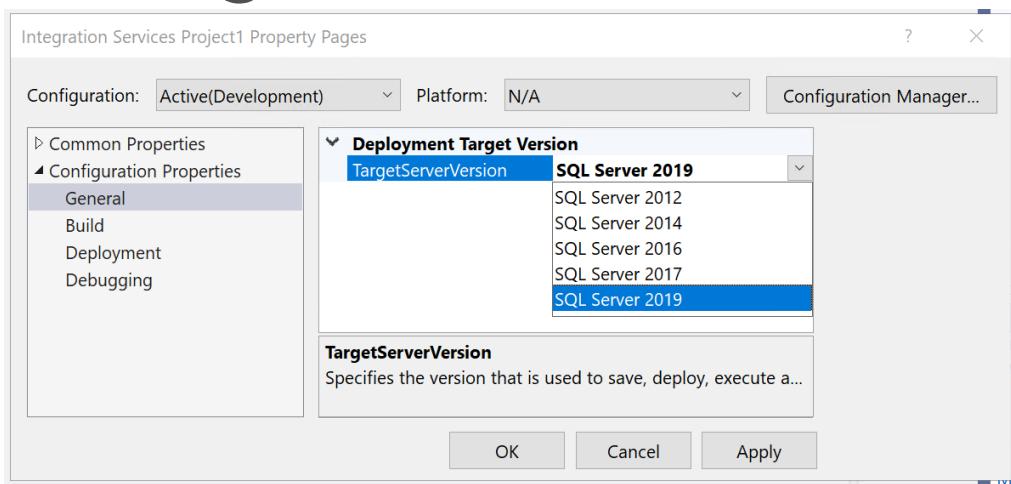
工具展開

The screenshot shows the Microsoft SQL Server Data Tools (SSDT) interface. On the left, the 'SSIS 工具箱' (SSIS Toolbox) is open, displaying a tree view of various SSIS components and tasks. A red box highlights this toolbox. In the center, the 'Package.dtsx [設計]' (Design) tab of the main editor window is active, showing the 'Control Flow' tab selected. Below it, the 'Connections Manager' and 'Variables' windows are visible. A large red arrow points from the toolbox towards the central design area. On the right, the 'Solution Explorer' window is open, showing the project structure for 'Integration Services Project2'. It includes items like 'Project.params', 'Integration Services Project2', 'SSIS 封裝' (Integration Services Package), 'Package.dtsx', and '連接的 Azure 資源' (Connected Azure Resource). Another red box highlights the Solution Explorer. At the bottom, a dark bar contains the text '變數' (Variables).

變數

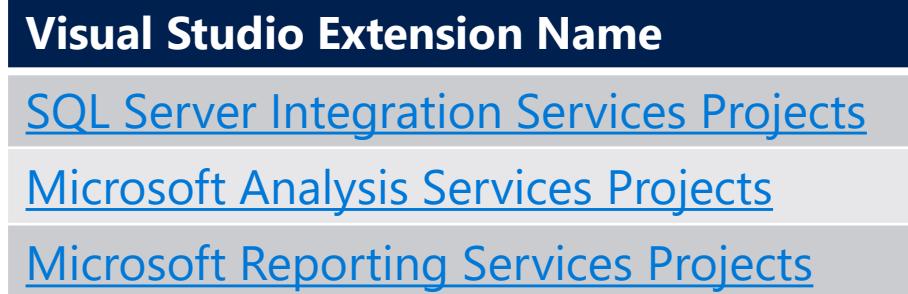
SSDT for Visual Studio 2017

- Standalone SSDT
 - Visual Studio 2017 bundled
- Can install VS project templates to existing VS instance
- Can target SSIS 2012 to 2019

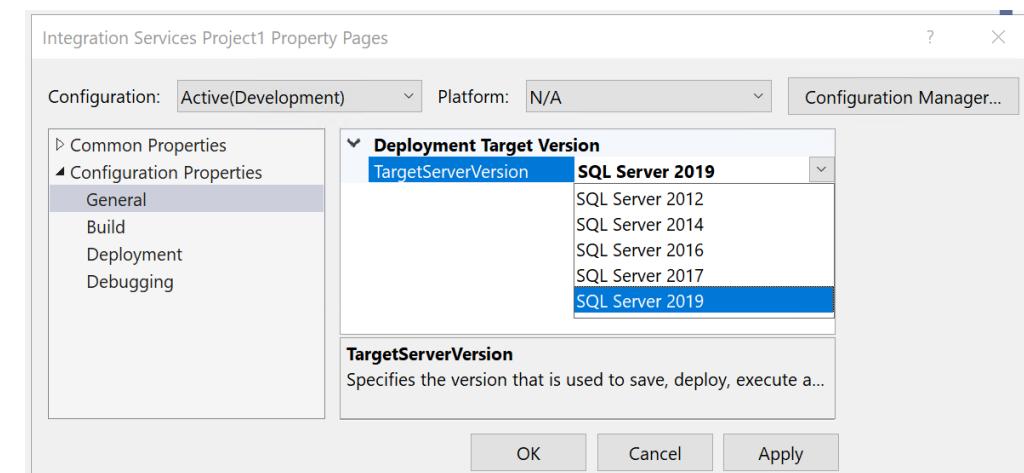
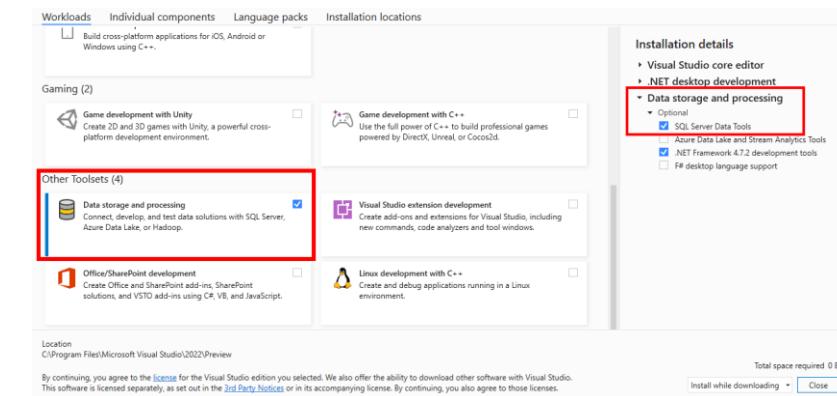


SSDT for Visual Studio 2019

- No more standalone SSDT installation
- Install Visual Studio Community Edition
 - SQL Server Data Tools (allows for creation of Database Project)
- Install Visual Studio Extensions



- Can Target SSIS 2012 to 2019



SSDT Walkthrough

Demo

SQL Server Management Studio (SSMS)

- Provides ability to deploy SSIS Catalog DB.
- Provides ability to schedule packages via SQL Server Agent.
- Configure and setup proxy accounts for execution.
- View current executions & history vis SSIS Catalog DB reports.
- Start/Stop Packages.

SSMS Walkthrough

Demo

dtutil.exe

用來管理 SQL Server Integration Services 封裝
可以複製、移動、刪除封裝，或確認封裝是否存在

- Command line tool used to manage packages
 - copy, move, delete, or verify the existence of a package
- Manage SSIS package stored in/on:
 - /DTS - package stored in SSIS package store
 - /SQL - package stored in msdb
 - /FILE - package stored on file system

```
dtutil /SQL srcPackage /COPY DTS;destFolder\destPackage
```

```
dtutil /FILE c:\sourcepkg.dtsx /DestServer <servername> /COPY SQL;destpkgname
```

```
for %%f in (*.dtsx) do dtutil /file %%f /En sql;SQLPDB\%%~nf;3;sql2005 /dec sql2005 /DestS .
```

dtutil.exe 32-bit vs 64-bit

- Use the 64-bit version

Version	Executable Location
64-bit	C:\Program Files\Microsoft SQL Server\150\DTS\Binn
32-bit	C:\ Program Files(x86)\Microsoft SQL Server\150\DTS\Binn C:\Program Files (x86)\Microsoft SQL Server Management Studio 18\Common7\IDE\CommonExtensions\Microsoft\SSIS\150\Bin n\

Administrator: Command Prompt

```
C:\Users\jaahmed>dtutil /?
Microsoft (R) SQL Server SSIS Package Utilities
Version 11.0.7001.0 for 32-bit
Copyright (C) Microsoft Corporation 2012. All rights reserved.

This applicat
ence Studio,
012 Standard,
stall a compo
C:\Users\jaahmed>
```

Administrator: Command Prompt

```
C:\Program Files\Microsoft SQL Server\150\DTS\Binn>dtutil /?
Microsoft (R) SQL Server SSIS Package Utilities
Version 15.0.2000.5 for 64-bit
Copyright (C) 2019 Microsoft. All rights reserved.
```

Usage: DTUtil /option [value] [/option [value]] ...
Options are case-insensitive.
A hyphen (-) may be used in place of a forward slash (/).
The vertical bar (|) is the OR operator and is used to list possible values.
For extended help use /help with an option. For example: DTUtil /help Copy

Command Prompt

```
C:\Program Files (x86)\Microsoft SQL Server Management Studio 18\Common7\IDE\CommonExtensions\Microsoft\SSIS\150\Binn>dtutil /?
Microsoft (R) SQL Server SSIS Package Utilities
Version 15.0.2000.118 for 32-bit
Copyright (C) 2019 Microsoft. All rights reserved.
```

Usage: DTUtil /option [value] [/option [value]] ...
Options are case-insensitive.
A hyphen (-) may be used in place of a forward slash (/).
The vertical bar (|) is the OR operator and is used to list possible values.
For extended help use /help with an option. For example: DTUtil /help Copy

/C[opy]	{SQL FILE DTS};Path
/Dec[rypt]	Password
/Del[ete]	
/DestP[assword]	Password
/DestS[erver]	Server
/DestU[ser]	User name
/DT[S]	PackagePath
/Dump	Process ID
/En[crypt]	{SQL FILE DTS};Path;ProtectionLevel[;Password]
/Ex[ists]	
/FC[reate]	{SQL DTS};ParentFolderPath;NewFolderName
/FDe[lete]	{SQL DTS};ParentFolderPath;FolderName
/FDi[rectory]	{SQL DTS}[;FolderPath[;S]]
/FE[xists]	{SQL DTS};FolderPath
/FR[ename]	{SQL DTS};ParentFolderPath;OldFolderName;NewFolderName
/Fi[le]	Filespec
/H[elp]	[Option]
/I[DRegenerate]	
/M[ove]	{SQL FILE DTS};Path
/Q[uiet]	
/R[emark]	[Text]
/Si[gns]	{SQL FILE DTS};Path;Hash
/SourceP[assword]	Password
/SourceS[erver]	Server
/SourceU[ser]	User name
/SQ[L]	PackagePath

dtexec 命令提示字元公用程式用來設定及執行 SQL Server Integration Services 套件

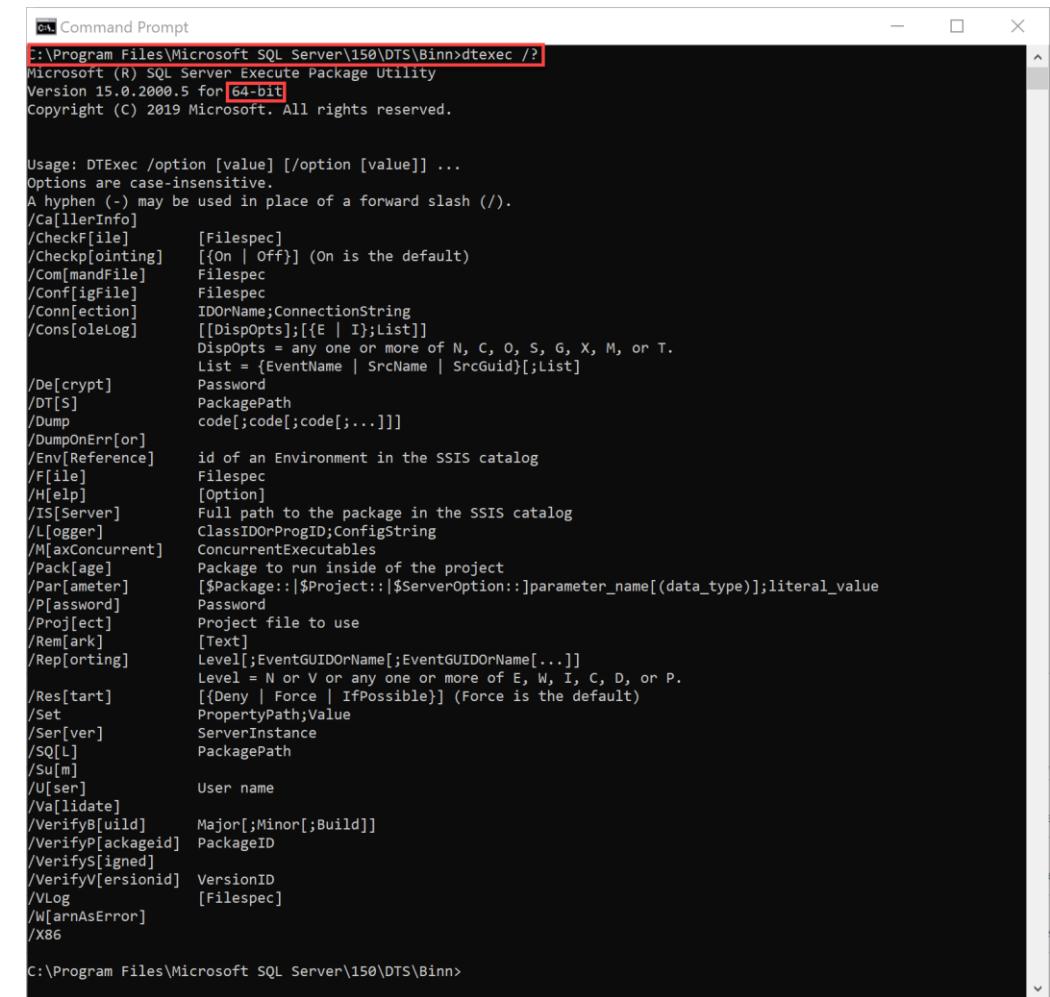
dtexec.exe

- Command line tool used to execute a package from
 - .ispac project file
 - .dtsx File system
 - Msdb
 - Ssisdb
- Command line arguments used to configure parameters, connections, properties, variables, logging for execution
- Temporarily upgrades older packages for execution (but not saved)
- “*SQL Server Integration Services Package*” SQL Job Step will call this executable to run the package

dtexec.exe 32-bit vs 64-bit

- to install 32-bit , you must install Client Tools or SQL Server Data Tools (SSDT)

Version	Executable Location
64-bit	C:\Program Files\Microsoft SQL Server\150\DTS\Binn\
32-bit	C:\Program Files (x86)\Microsoft SQL Server\150\DTS\Binn\ C:\Program Files (x86)\Microsoft SQL Server Management Studio 18\Common7\IDE\CommonExtensions\Microsoft\SSIS\150\Binn\



```
Command Prompt
D:\Program Files\Microsoft SQL Server\150\DTS\Binn>dtexec /?
Microsoft (R) SQL Server Execute Package Utility
Version 15.0.2000.5 for [64-bit]
Copyright (C) 2019 Microsoft. All rights reserved.

Usage: DTExec /option [value] [/option [value]] ...
Options are case-insensitive.
A hyphen (-) may be used in place of a forward slash (/).
/CallerInfo
/CheckFile [Filespec]
/CheckPointing [{On | Off}] (On is the default)
/CommandFile Filespec
/ConfigFile Filespec
/ConnectionString IDOrName;ConnectionString
[[DispOpts];[{E | I};List]]
DispOpts = any one or more of N, C, O, S, G, X, M, or T.
List = {EventName | SrcName | SrcGuid}[;List]
/ConnLog Password
/ConsoleLog PackagePath
/Decrypt code[;code[;code[;...]]]
/DT[S]
/Dump
/DumpOnError
/Env[Reference] id of an Environment in the SSIS catalog
/File Filespec
/Help [Option]
/IS[Server] Full path to the package in the SSIS catalog
/Logger ClassIDOrProgID;ConfigString
/M[axConcurrent] ConcurrentExecutables
/Pack[age] Package to run inside of the project
/Par[ameter] [$Package:::$Project:::$ServerOption:::]parameter_name[(data_type)];literal_value
/P[assword] Password
/Proj[ect] Project file to use
/Rem[ark] [Text]
/Rep[orting] Level[;EventGUIDOrName[;EventGUIDOrName[...]]]
Level = N or V or any one or more of E, W, I, C, D, or P.
[{Deny | Force | IfPossible}] (Force is the default)
/Res[tart] PropertyPath;Value
/Set ServerInstance
/SQ[L]
/Su[m] PackagePath
/U[ser] User name
/V[alidate]
/VerifyBuild Major[;Minor[;Build]]
/VerifyPackageID PackageID
/VerifySigned
/VerifyVersionID VersionID
/VLog [Filespec]
/W[arnAsError]
/X86

C:\Program Files\Microsoft SQL Server\150\DTS\Binn>
```

dtexec.exe - Exit Codes

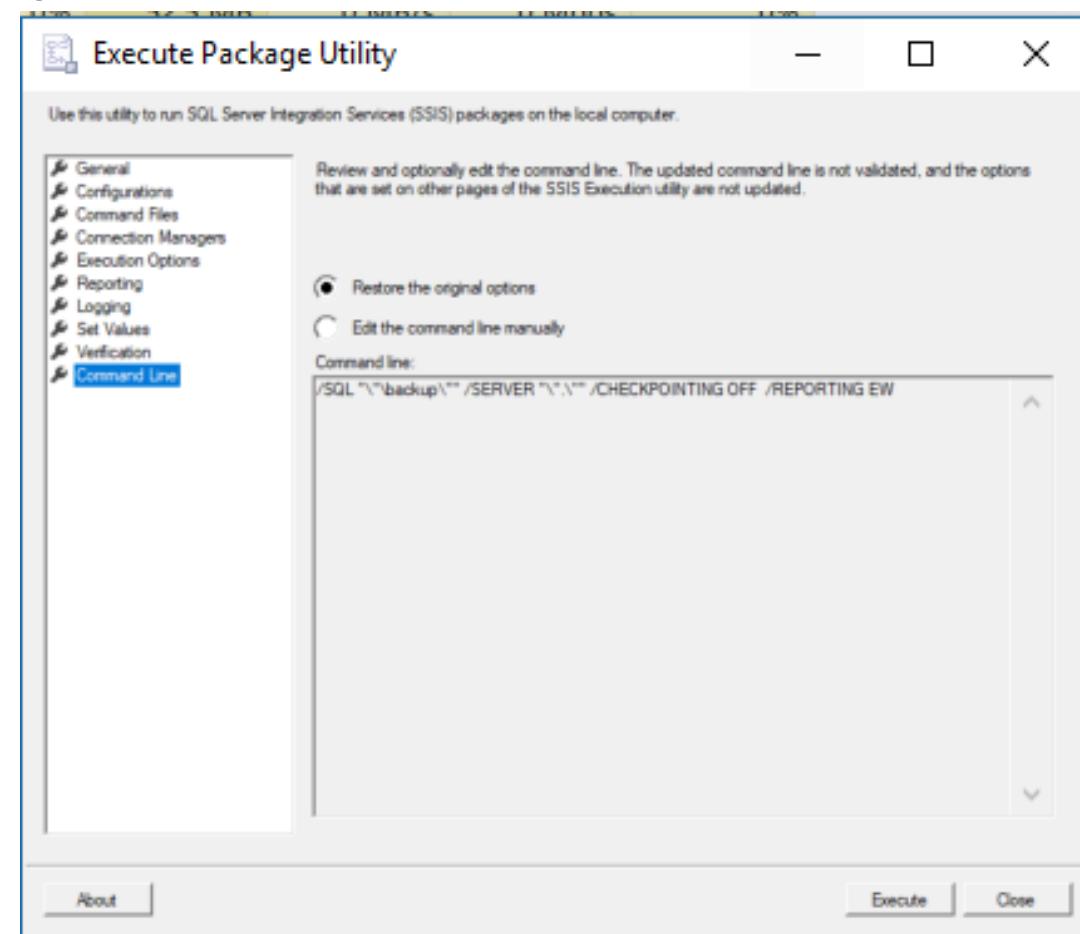
- Dtexec returns exit code
- exit code is used to populate the ERRORLEVEL variable

Exit Code Value	Description
0	The package executed successfully
1	The package failed .
3	The package was canceled by the user.
4	The utility was unable to locate the requested package. The package was not found .
5	The utility was unable to load the requested package. The package could not be loaded .
6	The utility encountered an internal error of syntactic or semantic errors in the command line.

dtexecUI.exe

- GUI to build command line argument for dtexec.exe
- can also invoke Utility is 32-bit only
- Execute button will start call will dtexec.exe 32-bit.

Version	Executable Location
32-bit	<i>C:\Program Files (x86)\Microsoft SQL Server Management Studio 18\Common7\IDE\CommonExtensions\Microsoft\SSIS\150\Binn\</i>



Getting to know SSDT

Lab 2

Questions?

- Can you run SSDT without Visual Studio?

No, it is an addon for Visual Studio. However you do not require full version of visual studio.

- Where can we see the status of SSIS packages on your server?

Look at SSIS Catalog DB Dashboard.

- How can we change version of the SSIS package?

On the properties of the project go to properties > target server version.



Module 3: SSIS Development

Agenda

- Project vs Package
- Connection Manager
- Variables vs Parameters
- Expressions
- Connection Strings
- Control Flow
- Precedence Constraints
- Data Flow
- Event Handlers
- Control Flow Parts

Project vs Package

Projects Contains

- One or more packages
- Project Parameters
- Connection managers
(shared connections)
- Control flow parts

Package Contains

- Control Flow
- Data Flow
- Package Parameters
- Event Handlers
- Connection manager
(dedicated).

Projects vs Packages

Demo

Connection Manager

- Allows SSIS to connect to any number of sources for both input and output.
- There are built in drivers such as OleDB, Flat File, ADO.NET, Analysis Service, etc.
- Additional connections drives such as Odata, Hadoop, and Azure has been added in SQL Server 2016+.
- Connections defined in connection manger are used for Control Flow, Data Flow, and Log Providers.

Connection Manager (2)

- Following connection managers are only available in Control Flow.
 - FTP, HTTP, MSMQ, SMOServer, SMTP, and WMI

Connection Manager (3)

- Following connection managers are available to both Source and Destination data flows:
 - Database (ADO.NET, OLE DB, or CDC Source) or File (Excel, Flat File, XML, Raw File).
- Following connection managers are available only to Destination data flows:
 - SSAS (Data mining model training, Dimension processing, or partition processing) or Rowset (Data reader, Recordset).

Connection Manager

Demo

Variables vs Parameters

Variables

- Are package level only.
- Are read-write.

Parameters

- Are package or project level.
- Cannot be modified directly but can be updated as a pass-through value.
- Are type of variable.

Variables vs Parameters – Supported Data Types

Variables

- Types available:
 - All the types of parameters.
 - Plus Char, DBNULL, Objects.

Parameters

- Types available:
 - Boolean, Byte, DateTime, Decimal, Double, Int16, Int32, Int64, Sbyte, Single, String, UInt32, and UInt64.

Variables vs Parameters – Scope

Variables

- Defined at development, can be granular as task level.
- Can change the variable in design time only.
- Scope: Package Level, Container Level, Data Flow, and Task Level.
- Visible as `User::VariableName`

Parameters

- Project Level – Visible to all Packages.
 - Visible as `$Project::ParameterName`
- Package Level – Visible to package defined in.
 - Visible as `$Package::ParameterName`

Variables vs Parameters – System

Variables

- System variable available for Projects, Packages, Tasks, etc. and are read-only.
 - Visible as `System::VariableName`
 - All system variables are not visible in all scopes (e.g. `System::ErrorCode` and `System::ErrorDescription` are only visible in `OnError` event handler).

Parameters

- No system parameters.

Variables vs Parameters – Additional Information

Variables

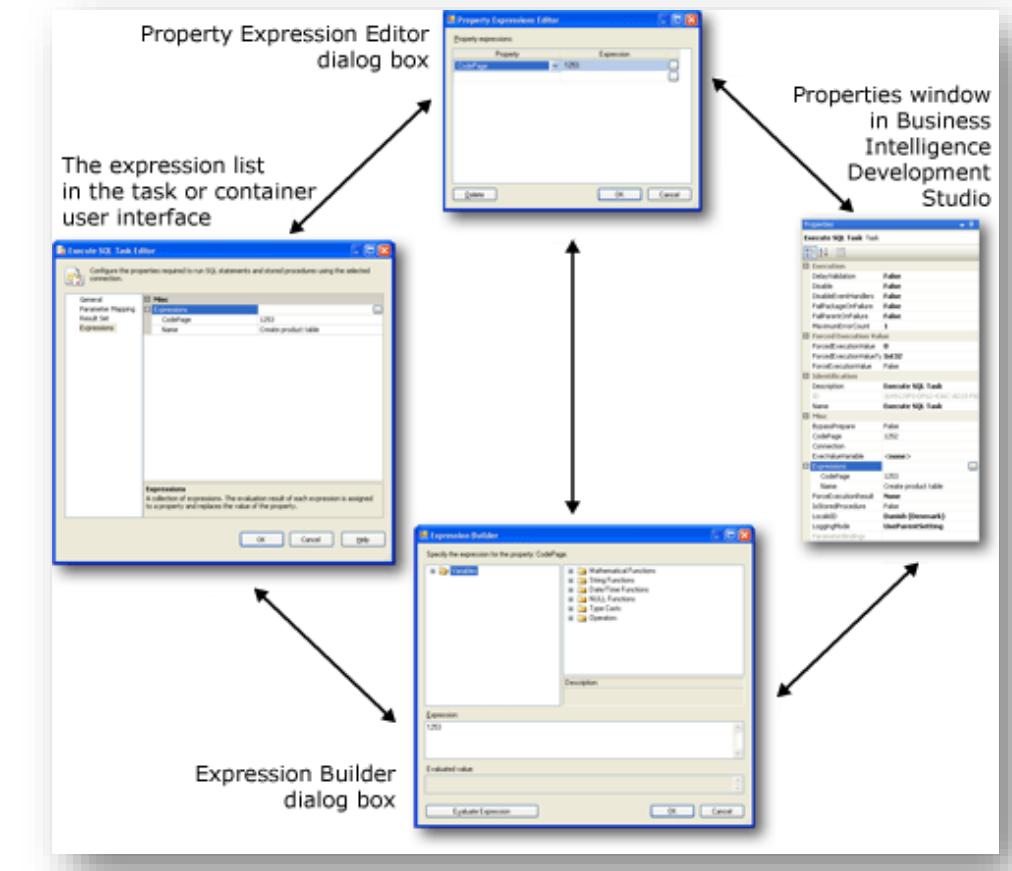
- Some variables need to be saved in the database to allow for persistence (e.g., CDC Control Task).

Parameters

- Order of precedence for value assignment is Execution > Server > Design.
- Has additional properties on sensitive and required.

Expressions

- Used to dynamic assignment or comparison, instead of hard-coding values.
- Expression syntax is like C# Style
 - Value Assignment uses single equal (=)
 - Equality Comparison uses double equal (==)
 - Precedence Constraint, Conditional Split Transform
 - Data Type Casting : (data_type_to_cast_to) value
 - Variable referenced as
 @ [NameSpace::VariableName] eg:
 @ [User::MyCounter]

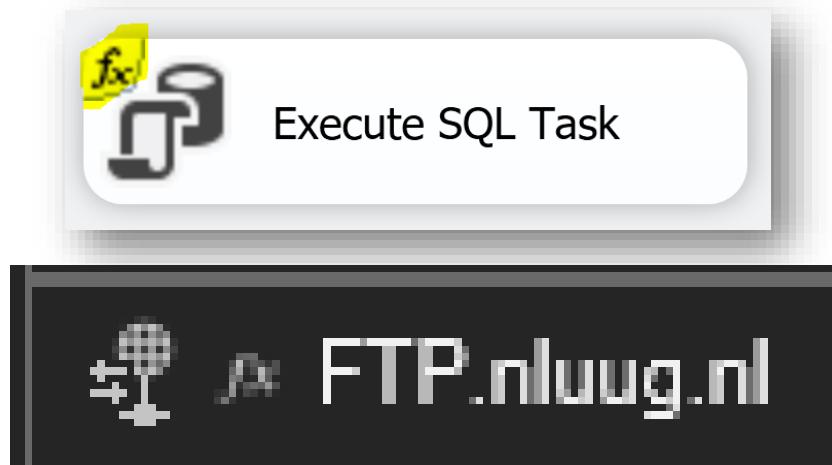


Expressions – Additional Info

- Expression Builder GUI – Lists functions & function input parameters
 - New SQL 2012+ expression functions – LEFT, REPLACENULL, TOKEN, TOKENCOUNT
 - Removal of 4000-character limit for expressions, in SQL 2012+.

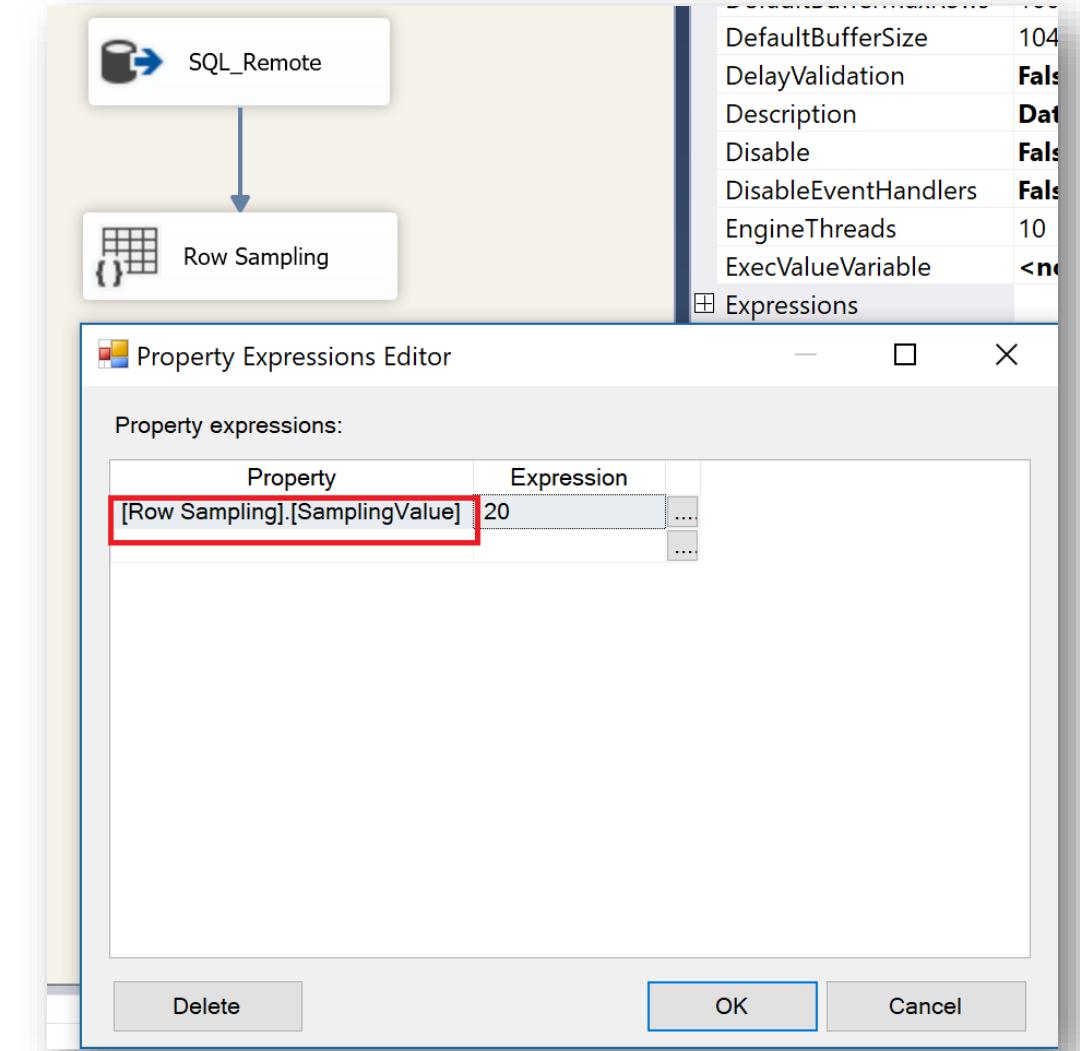
Property Expressions

- Dynamically update task property at run time.
- Any task or object that has expressions has “fx” added to the object to signal it is using expressions.
- Can be set for:
 - Package, Task, Foreach Loop, For Loop, Sequence, Foreach enumerator, Event Handler, A package or project level connection manager, or Log Provider.



Property Expressions – Data Flow

- Data Flow expressions are not straight forward. To set the expression, you must set the expression at Data Flow Task in Control Flow.
- Expressions are not available for all data flow tasks.



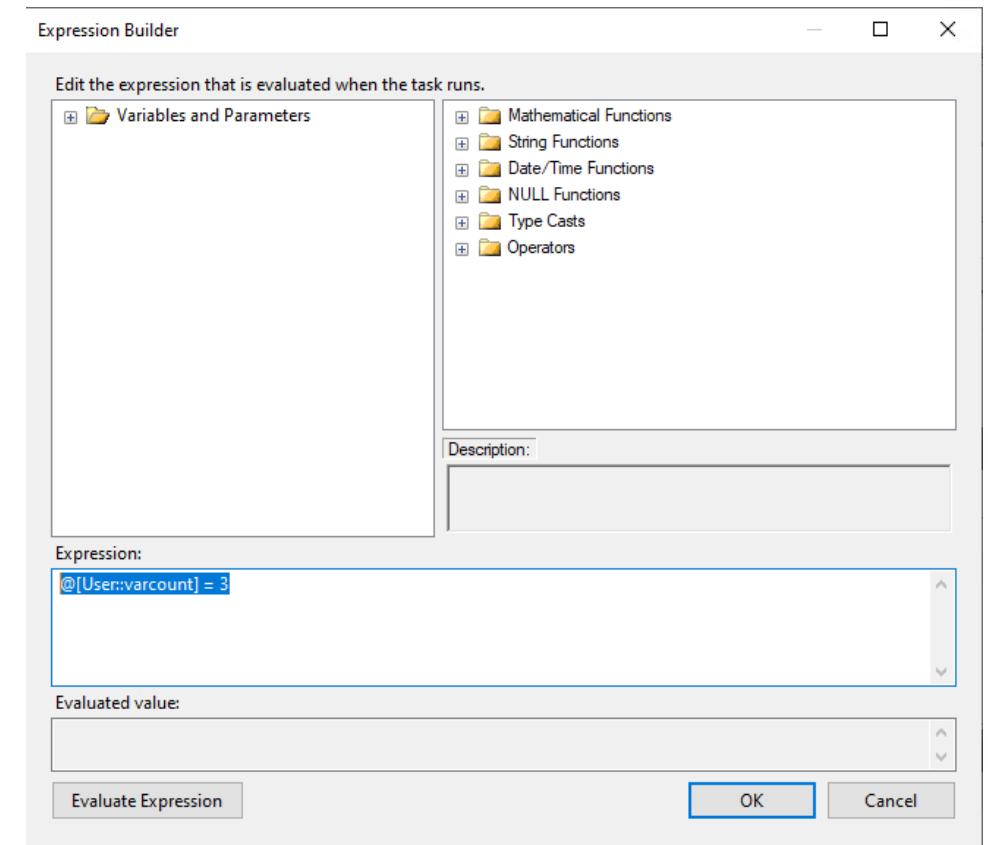
Variable Expressions

- Assigns value to variable.
- Dynamically update variable value property at run time.
- Expression markers.

Variables					
	Name	Scope	Data ...	Value	Expression
	i	Package1	Int32	911	
	VarExpression	Package1	Int32	911	@[User::i]

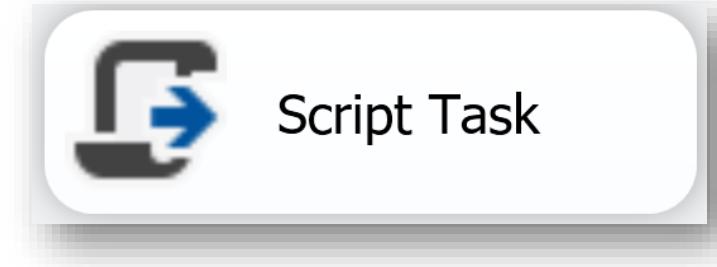
Expression Task

- Assigns value to only one variable.
- Introduced in SQL Server 2012+.
- Not available in Data Flow Task.



Control Flow Script Task

- Can also be used to assign value to variable(s).
- .NET script can be used to assign values
 - Example:
Dts.Variables["User::MyIntVar"].Value
=911



```
C#  
  
using System;  
using System.Data;  
using Microsoft.SqlServer.Dts.Runtime;  
  
public class ScriptMain  
{  
  
    public void Main()  
    {  
        int customerCount;  
        int maxRecordCount;  
  
        if (Dts.Variables.Contains("CustomerCount") == true && Dts.Variables.Contains("MaxRecordCount") == true)  
        {  
            customerCount = (int) Dts.Variables["CustomerCount"].Value;  
            maxRecordCount = (int) Dts.Variables["MaxRecordCount"].Value;  
  
        }  
  
        if (customerCount > maxRecordCount)  
        {  
            Dts.TaskResult = (int)ScriptResults.Failure;  
        }  
        else  
        {  
            Dts.TaskResult = (int)ScriptResults.Success;  
        }  
    }  
}
```

Expression Task vs Script Task

Variable Assignment

Expression Task

- Only one value assignment
- Quicker to modify

Script Task

- Multiple value assignment possible.
- A bit longer to modify, VSTA (VSTA need to be loaded each time to modify).

Script Task Example

- <https://learn.microsoft.com/en-us/sql/integration-services/extending-packages-scripting-task-examples/detecting-an-empty-flat-file-with-the-script-task?view=sql-server-ver16>
- <https://learn.microsoft.com/en-us/sql/integration-services/extending-packages-scripting-task-examples/script-task-examples?view=sql-server-ver16>

Parameters & Variables & Expressions

Demo

Sensitive Data

- Sensitive SSIS properties
 - Password for connection string
- Not all properties are sensitive, for example, we might want to protect Username also.
 - Workaround is promote the parameter for username to sensitive.

Properties	
OLEDB.MOGUPTA-PC01.AdventureWork	
ConnectionString	Data Source=MOG
DataSourceID	
DelayValidation	False
Description	
Expressions	
HasExpressions	False
ID	{74C20F86-AC7B-4F
InitialCatalog	AdventureWorks201
Name	OLEDB.MOGUPTA-P
PackagePath	\Project.Connection
Password	*****
Qualifier	
RetainSameConnecti	False
ServerName	MOGUPTA-PC01
SupportsDTCTransac	True
UserName	

Connection Strings

- Defines how to connect to data source.
- Provides all the important information, for example, for connecting to SQL Server:
 - Server Name
 - Instance Name
 - Port
 - Authentication Type (Trusted vs User Name / Password)
 - Database Name

Connection Strings - Security

- Deploy packages to SSISDB Catalog
 - Catalog automatically encrypts/decrypts based on Sensitive property.
- Username is not sensitive
 - Workaround promote to property to parameter and set parameter as sensitive.
- Password is sensitive

Connecting String – Security (2)

Package Protection Level

- How to store sensitive values in file content
- Set Project and all Package ProtectionLevel to be same.

Package Protection Level	Notes
DontSaveSensitive	Sensitive data is not saved in package. Must supply on execution via parameters.
Encrypt Sensitive With Password	Password used to encrypt entire package content. To run or open in SSDT, must supply password.
Encrypt Sensitive With UserKey (default)	Uses current user profile to encrypt only sensitive values. Only same users can open package in SSDT. If different user opens in SSDT, must re-supply sensitive values.
Encrypt All With Password	Encrypt all values, not just sensitive with password.
Encrypt All With UserKey	Encrypt all values, not just sensitive with current user profile.

Connecting String – Security (3)

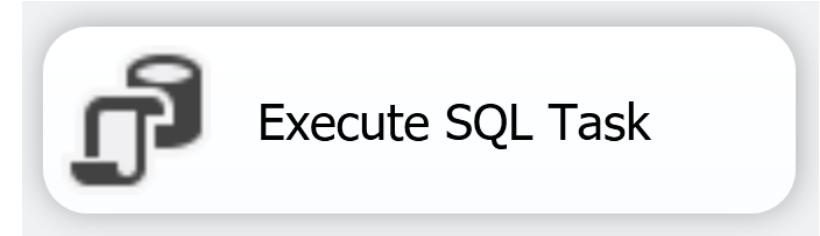
Package Protection Level	Notes
ServerStorage (does not show in dropdown)	<p>Used by SSISDB catalog, SQL 2012+</p> <ul style="list-style-type: none">Catalog automatically encrypts the package data and sensitive values.Projects Exported from catalog automatically set to EncryptSensitiveWithUserKeyWith each project deployment creates<ul style="list-style-type: none">Certificate - MS_Cert_Proj_<Project_ID>Symmetric Key - MS_Enckey_Proj_<Project_ID> which is used to encrypt the project.First project execution creates<ul style="list-style-type: none">Certificate - MS_Cert_Proj_Param_<Project_ID>Symmetric Key - MS_Enckey_Proj_Param_<Project_ID> which is used to encrypt the project parameter values.

Connection String

Demo

Execute SQL Task

- Used to execute SQL statement(s) on various connect managers EXCEL,OLE DB,ODBC,ADO,ADO.NET,SQLMOBILE.
E.g.: truncate table before calling data flow task
retrieve email list to send personalized notification.
- Required Properties
 - Connection Manager to use
 - SQL Statement to execute
- Can accept Input & Output parameters
 - Parameter Marker and Parameter Name differs by Connection Manager Type.
- Can assign result sets to variable



Execute SQL Task (2)

- Passing parameter values depend on the driver type.
- To capture an output in a result, add keyword OUTPUT in the SQL code.

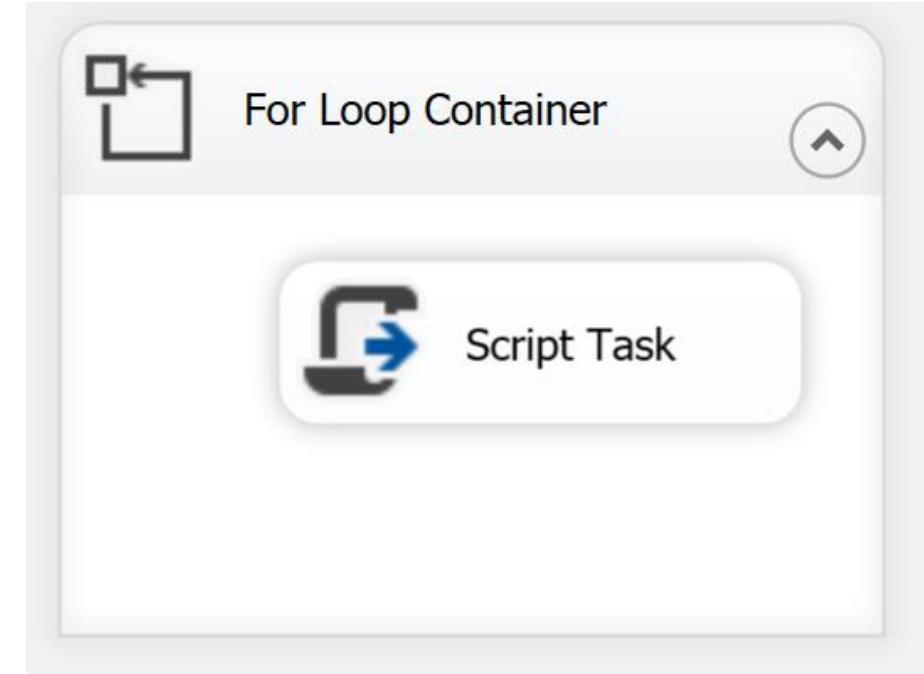
Connection Type	Marker	Example
ADO	?	SELECT * FROM T1 WHERE ID = ?
ADO.NET	@P1	SELECT * FROM T1 WHERE ID = @P1
SQLMOBILE	@P1	SELECT * FROM T1 WHERE ID = @P1
OLEDB	?	SELECT * FROM T1 WHERE ID = ?
ODBC	?	SELECT * FROM T1 WHERE ID = ?
EXCEL	?	SELECT * FROM T1 WHERE ID = ?

Using Parameters to Call Stored Procedure

Demo

For Loop Container

- Loop and execute tasks within container until expression evaluates too false.
- Required Properties
 - Must use variable
 - InitialExpression
 - EvalExpression
 - AssignExpression
 - Eg:



Property	Sample Setting to execute Tasks within container 4 times
InitialExpression	@[User::Counter]=1
EvalExpression	@[User::Counter]<=4
AssignExpression	@[User::Counter]=@[User::Counter]+1

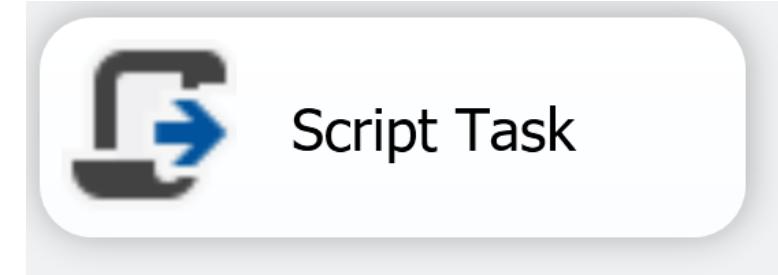
For Each Loop Container

- To Implement While Loop
- Declare variable , eg User::ExitFlag = False, Boolean
- Set User::ExitFlag=true within For Loop container when condition is met

Property	Settings
InitialExpression	@[User::ExitFlag]=false
EvalExpression	@[User::ExitFlag]=true

Script Task

- Allows to perform functions not available by other control tasks.
- Script can be written in C# or VB.Net
 - Can reference large collection of .Net library
- Script can read SSIS variable & parameters
- Script can write back to SSIS variables
- Also allows you to reuse pre-developed .NET Assemblies

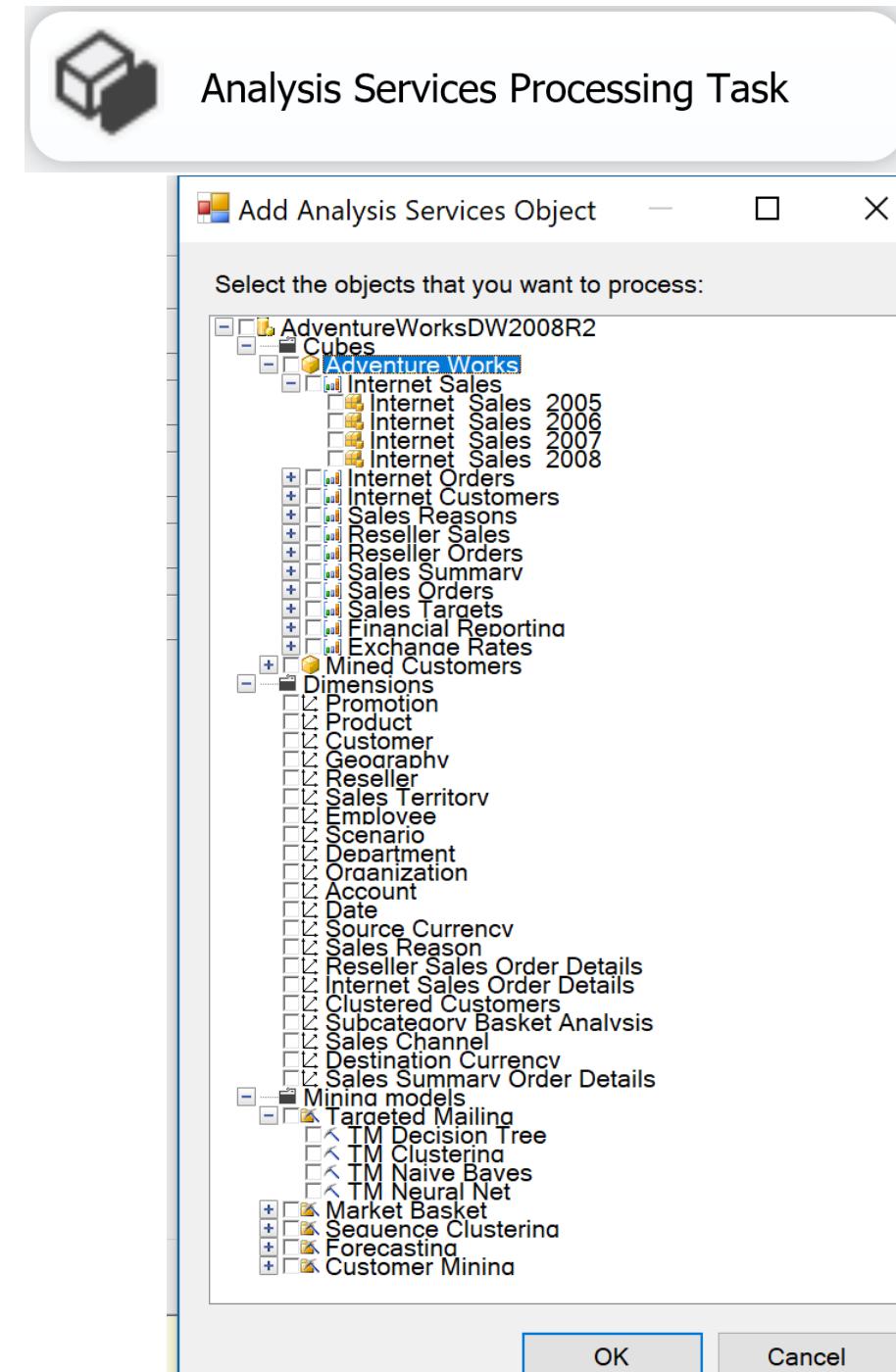


Using SQL to Get the Information

Demo

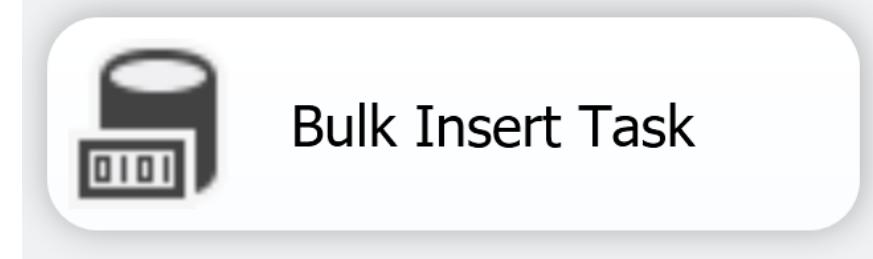
Analysis Services Processing Task

- Used to process Multidimensional and Tabular Analysis databases
- Required Properties
 - Connection Manager to use
 - What to process
 - Multidimensional - entire db, cube, measure, partition, dimension, mining model
 - Tabular – Table or Table Partition



Bulk Insert Task

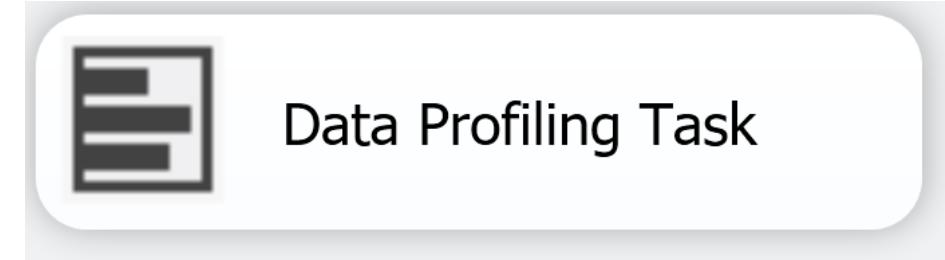
- Used to load data directly from file into SQL Server.
- Data can not be transformed while loading data into SQL Server
- Required Properties
 - Source Connection - Data file to load
 - Format File of the Data
 - Destination Connection Manager
 - Destination Table to load the data into.



Bulk Insert Task

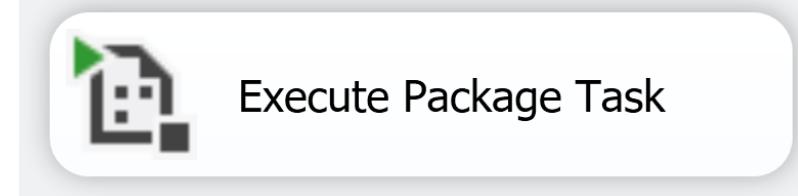
Data Profiling Task

- Profiles a data source and identifies data quality problems.
- Required Properties
 - DestinationType – File or Variable
 - Destination – FileConnection or VariableName
 - Select Profile Type Request(s)
 - Connection Manager
 - Table or View to profile
 - Column(s) to profile from Table or View
- File Destination preferred, to use the Data Profile Viewer



Execute Package Task

- Runs another package as part of current package's control flow.
- Required Properties
 - `PackageNameFromProjectReference` – Child package to execute
- Used for
 - Breaking down complex package workflow
 - Reusing parts of packages
 - Modular Package development, for multiple developers
- Also supports passing Parameters from Parent Package to Child Package



Execute Package Task

Execute Process Task

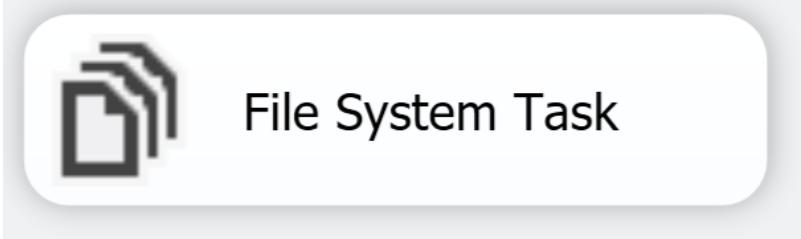
- runs an application or batch file
- Required Properties
 - Executable to run
- Used for
 - Calling external utilities
 - Reusing existing batch files, PowerShell scripts
 - Performing functionality unavailable in existing Control Flow Task
 - Eg: Secure FTP Transfer, File Compression/Uncompression
- Can also supply command line arguments



Execute Process Task

File System Task

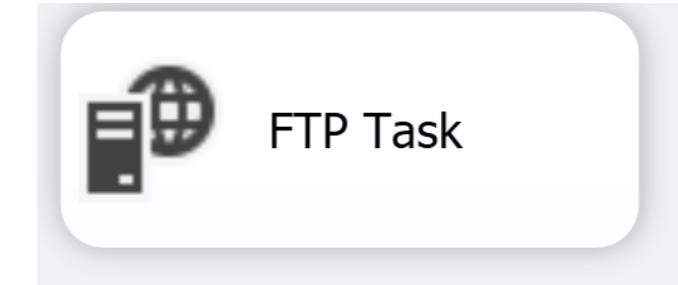
- Perform File and Directory operations in the file system.
- Required Properties
 - One or more File Connection Manager depending on operation
- Used to
 - Create/Copy/Delete/Move Directory
 - Copy/Delete/Move/Rename File
 - Set Attributes
 - Delete Directory contents
- Option to overwrite file available.
 - Will Error, if file exists and attempt to create/move file.



File System Task

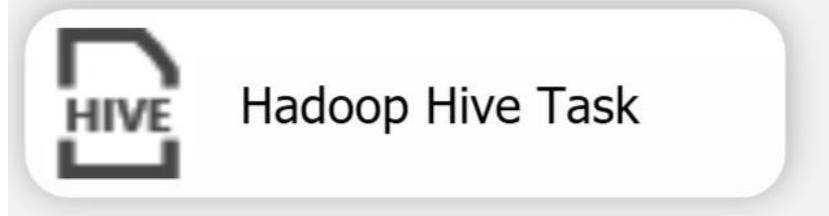
FTP Task

- Send email notifications
- Supports
 - File Attachments
 - Priority – Low, Normal, High
- Required Properties
 - FTP Connection Manager
 - Rest varies depend on action
 - For files transfer require remote path and Local File connection manager
- Used to
 - Download files from FTP server
 - Upload files to FTP Server
 - Create/remove folder on FTP Server
 - Create/remove folder on local Server
- Can not connect to Secure FTP (SFTP).



Hadoop Hive Task

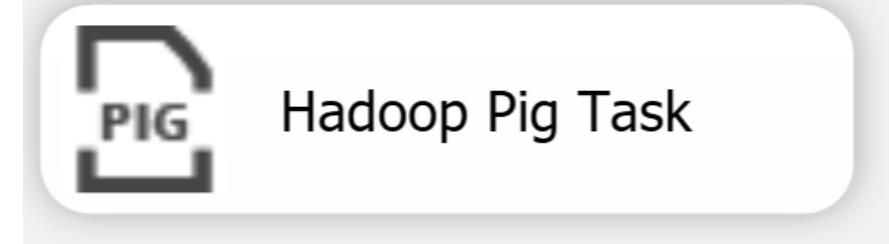
- Run Hive script on a Hadoop cluster
- Required Properties
 - Hadoop Connection Manager
 - InlineScript – Hive job script to execute



Hadoop Hive Task

Hadoop Pig Task

- Run Pig script on a Hadoop cluster
- Required Properties
 - Hadoop Connection Manager
 - InlineScript – Pig job script to execute



Hadoop File System Task

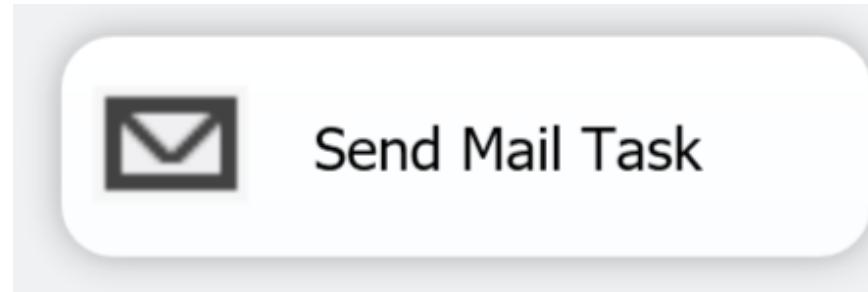
- Copy files from, to, or within a Hadoop cluster
- Required Properties
 - Hadoop Connection Manager
 - File Connection Manager (if copying from or to Hadoop)



Hadoop File System Task

Send Mail Task

- Send Email Notifications.
- Required Properties
 - SMTP Connection Manager
 - From email address
 - To email address
 - Subject
- Also supports
 - File attachments
 - Message Priority – Low, Normal, High



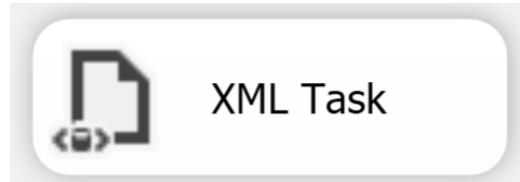
Web Service Task

- Execute a web service and store return value in file or variable.
- Required Properties
 - HTTP Connection Manager
 - WSDL file – download location for Web Services Description Language (WSDL)
 - Input – Web Service, Method, and input parameters
 - Output – File or variable to store returned value
- Allows developer to easily use vendors webservices



XML Task

- work with xml data.
- Required Properties
 - File Connection Manager(s)
 - Rest varies depend on XML Operation
- Used to
 - Transform XML using Extensible Stylesheet Language Transformations (XSLT)
 - Performs XPath queries and evaluations
 - Validate XML
 - Compare and Merge multiple xml documents



XML Task

XML Task (2)

XML Operation Type	Notes
Diff	Compare two XML documents, and write difference to XML Diffgram document.
Patch	Applies the output from the Diff operation, called a Diffgram document, to an XML document, to create a new parent document that includes content from the Diffgram document.
Validate	Validate XML document against Document Type Definition (DTD) or XML Schema definition (XSD) schema.
XPath	Performs XPath queries and evaluations.
XSLT	Transform XML documents using XSLT (Extensible Stylesheet Language Transformations)

For Each Loop Container

- Loop through specified enumerator.
- Similar to For Loop container.
- Required Properties
 - Enumerators
 - Additional properties specific to enumerator
- Available enumerators are

[Foreach File Enumerator](#)

[Foreach Item Enumerator](#)

[Foreach ADO Enumerator](#)

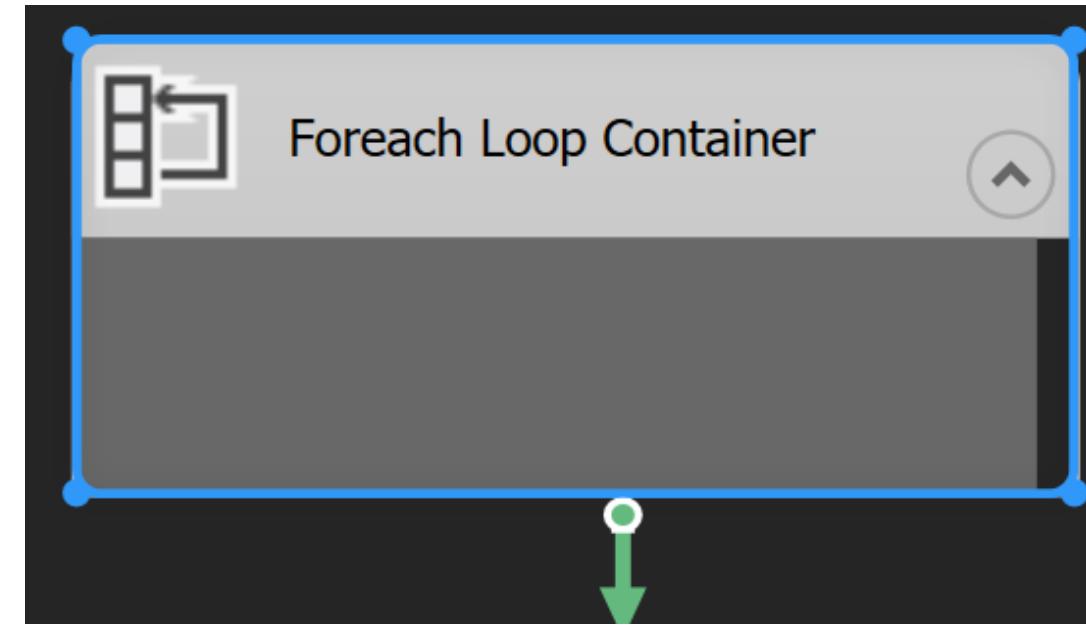
[Foreach ADO.NET Schema Rowset Enumerator](#)

[Foreach From Variable Enumerator](#)

[Foreach NodeList Enumerator](#)

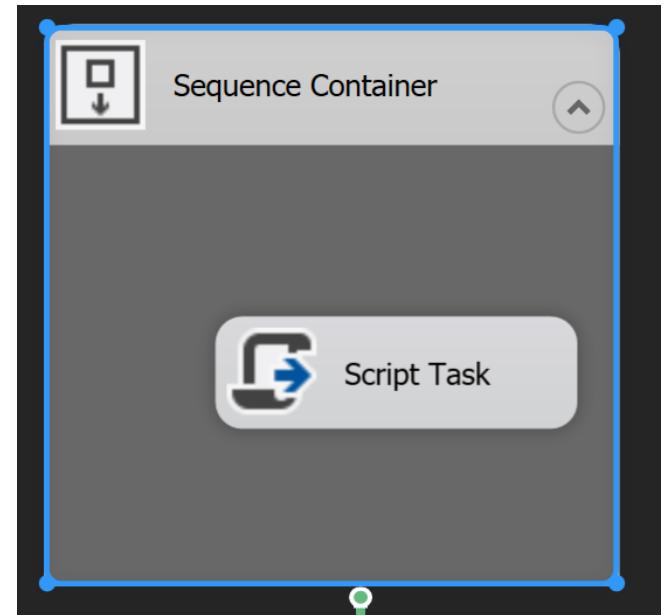
[Foreach SMO Enumerator](#)

[Foreach HDFS File Enumerator](#)



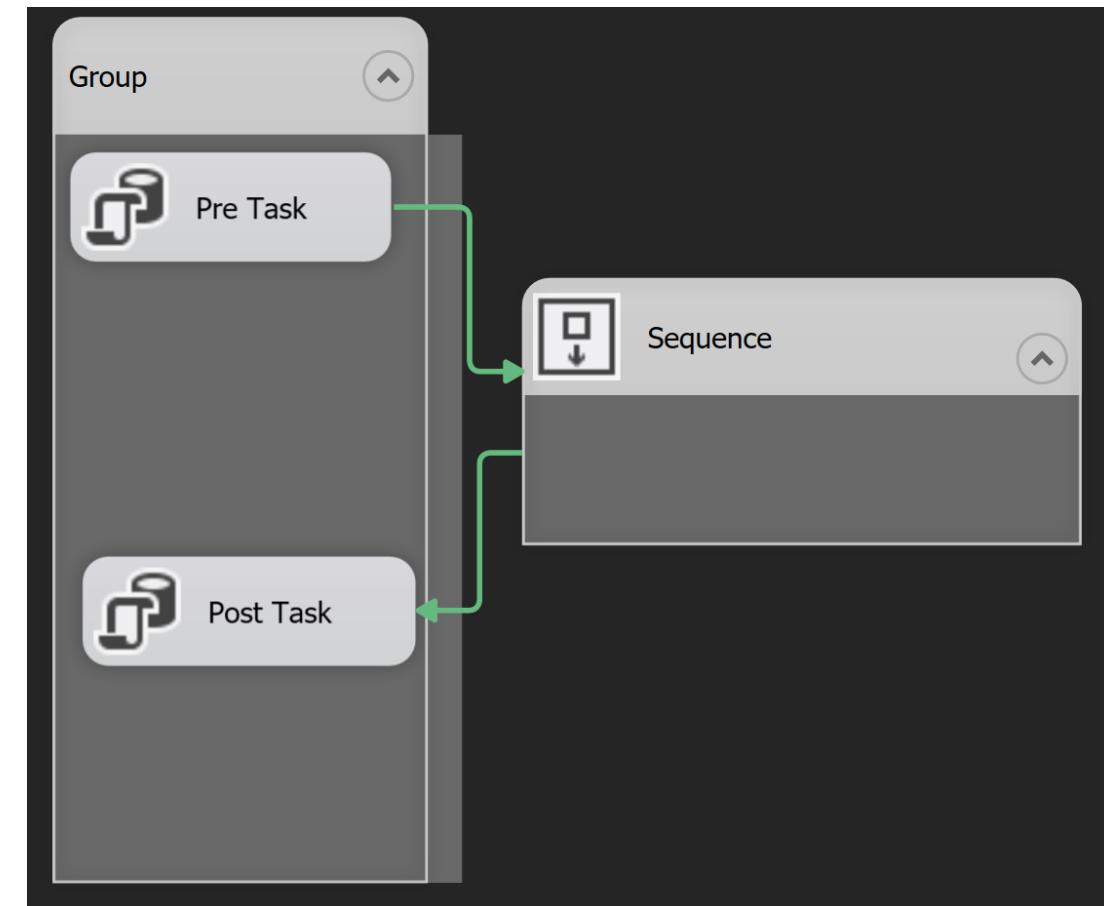
Sequence Container

- Group tasks into a container.
- Can be enabled/disabled
- Can participate in a transaction
- Only for grouping control flow tasks/containers



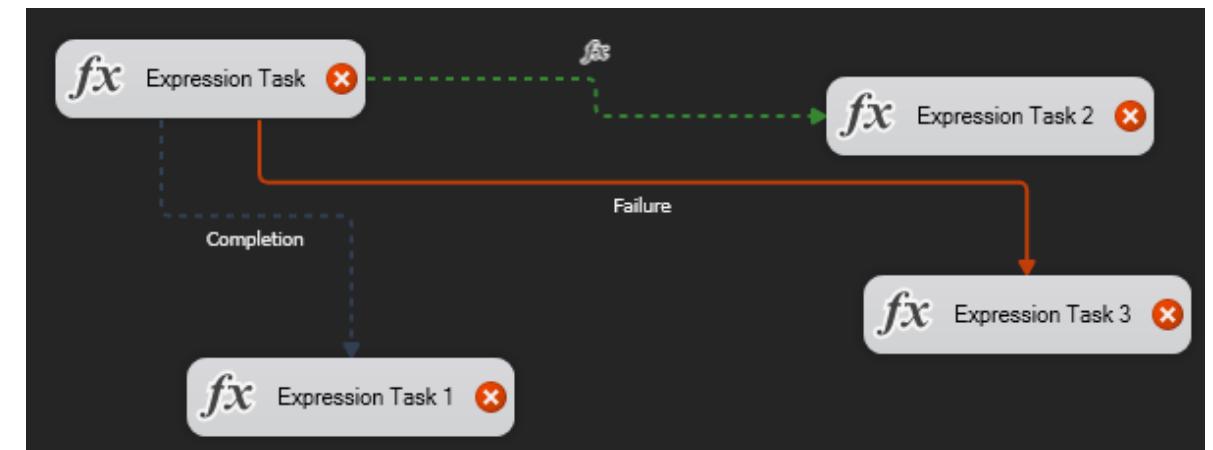
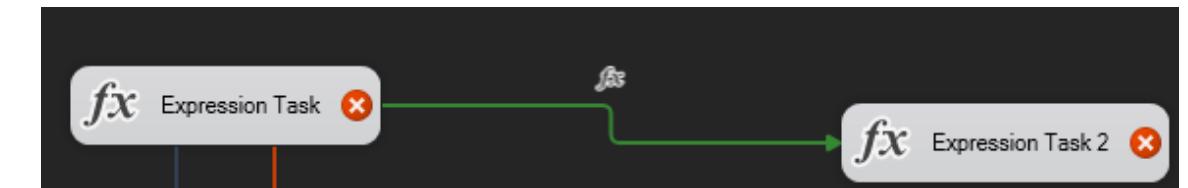
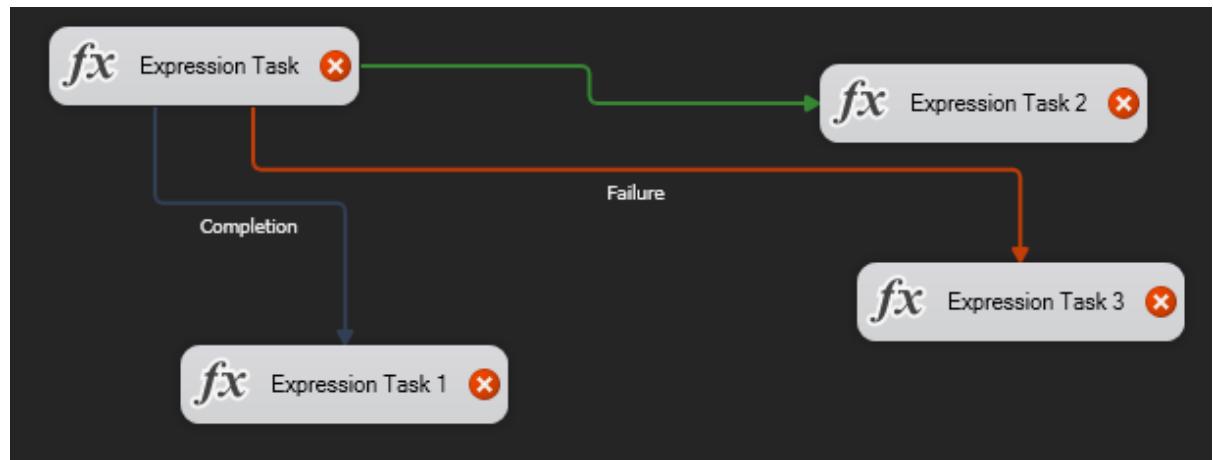
Grouping/Ungrouping components

- Group tasks within both Control flow and Data Flow
- Can not be enabled/disabled
- Can not participate in a transaction



Precedence Constraints

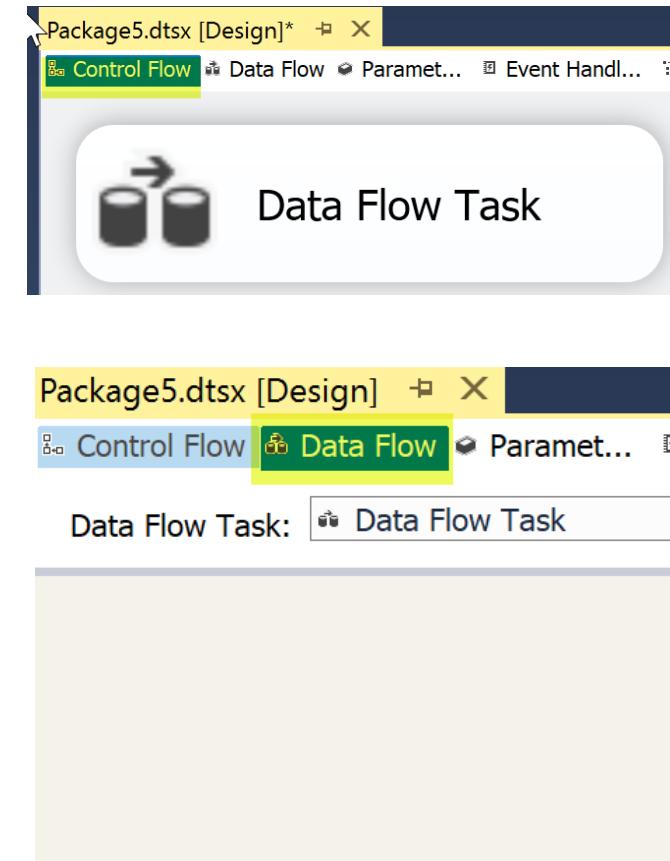
- Three Types
 - Success
 - Failure
 - Completion
- Can also make it on condition by using expressions.
- Can combine and/or logic.



Bringing it Together

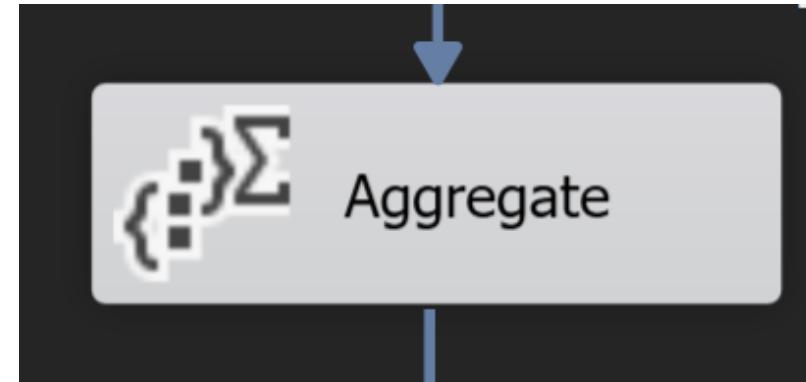
Data Flow Task

- Used to
 - Extract data from Source
 - Transform (clean/modify) data as it is moved from source to target
 - Load transformed data to Destination
- E.g.: Importing and Exporting data.
- Added on Control Flow tab, Designed in Data Flow tab
- No required properties but will do nothing by itself.
- This task will be covered in later module.



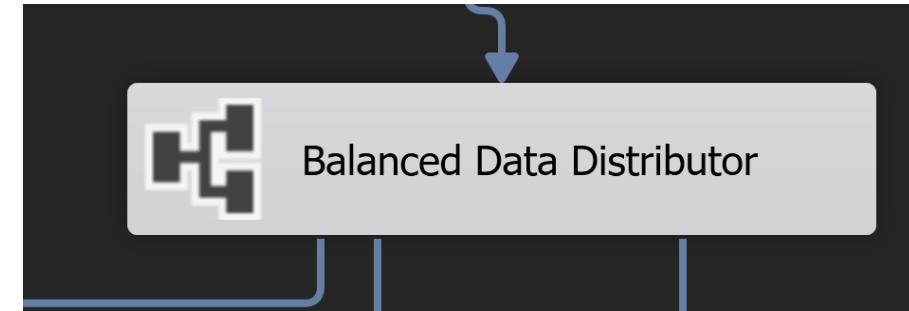
Aggregate Transform

- Performs Aggregate functions
 - Sum, Average, Min, Max, Count, Count Distinct
 - Group By
 - KeyScale may be required for large data sets
- Required Properties
 - Column to perform Aggregate
 - Type of Aggregate Operation to perform



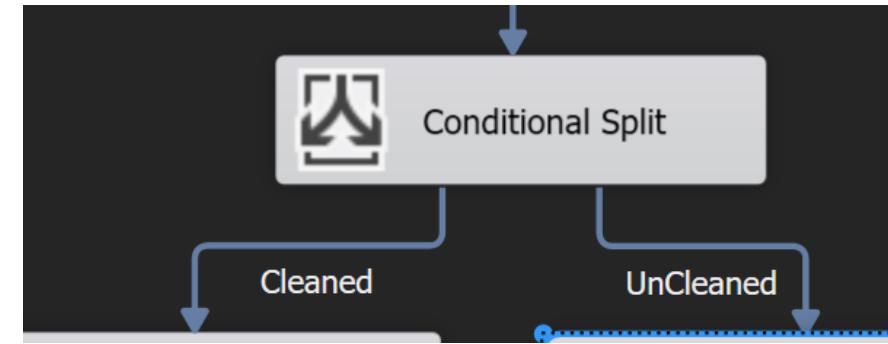
Balance Data Distributor

- Distributes buffers of incoming rows uniformly across outputs on separate threads
- Distributes data in round-robin fashion
- Distribution depends on Data Flow settings
 - [DefaultBufferSize](#)
 - [DefaultBufferMaxRows](#)



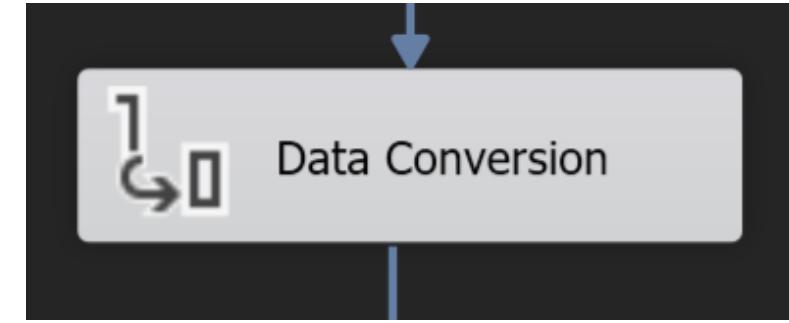
Conditional Split Transform

- Split the incoming data into different outputs based on Expression conditions
- Required properties
 - Expressions to split out data streams



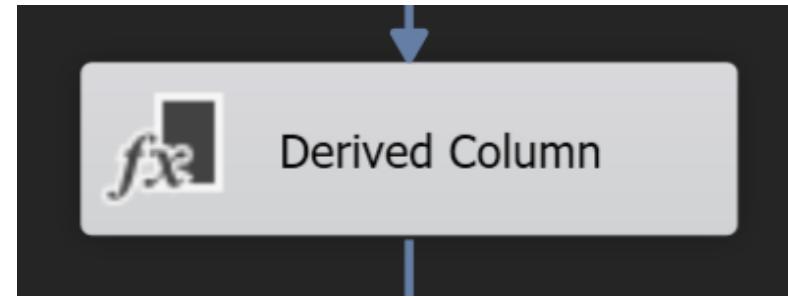
Data Conversion Transform

- Converts data from one data type to another.
- Required properties
 - Column to convert the data type to
 - New Data Type
 - New Column Name
- Similar operation can also be performed by Derived Columns



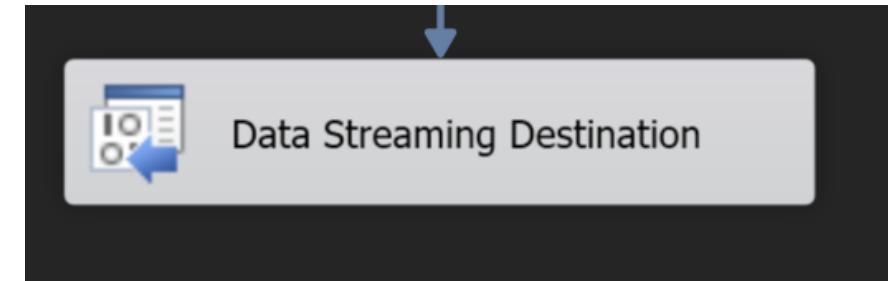
Derived Column Transform

- creates new column by applying transformation expressions to input.
- Required properties
 - Column Name – New or Replace
 - Column Transformation Expression
- Can achieve same functionality as Data Conversion
 - by performing Type Casting Expression transformation you are achieving the same functionality as Derived Columns



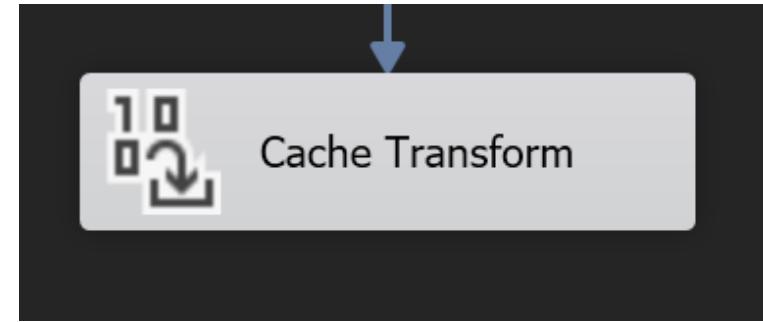
Data Streaming Destination

- Exposes data via SSISOLEDB Provider
- Allows you to create a linked server to query results from SSIS Package execution
 - Use the Data Feed Publishing Wizard
 - Select SSIS Package
 - Select Database
 - View Name which will query linked server that returns results from executing SSIS package
 - Linked Server Name



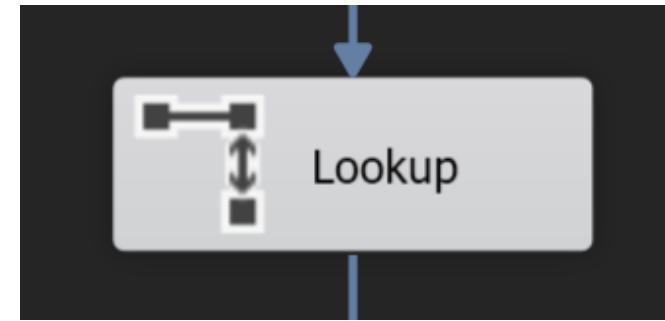
Cache Transform

- Writes data to a cache (a .caw file) with Cache Connection Manager.
- Reuse cache between multiple Lookup transforms to reduce database load.
- Share the case to reduce Memory usage.
- cache data stored in clear text not recommended for sensitive data.
- Requirements
 - Cache Connection Manager
 - Must set Index Position



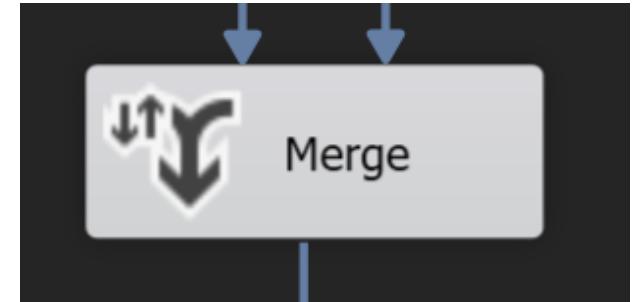
Lookup Transform

- Joins additional columns to data flow by looking up values in a table.
- Required Properties
 - OLEDB Connection Manager or Cache Connection Manager
 - Lookup Table Name
 - Column Mapping for Joining
 - Lookup Table's Column Name to Output



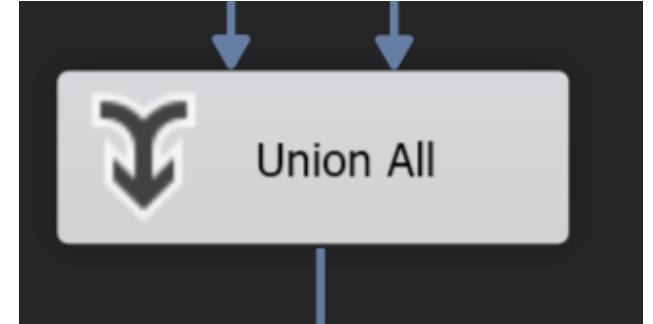
Merge Transform

- Combines rows from multiple sorted data flows into one sorted data flow. (ie. Append)
- Required Properties
 - Pre **Sorted** inputs



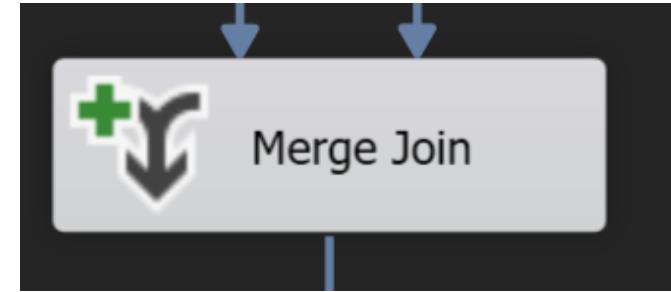
Union All Transform

- Combines rows from multiple data flows into one data flow. (ie. Append)



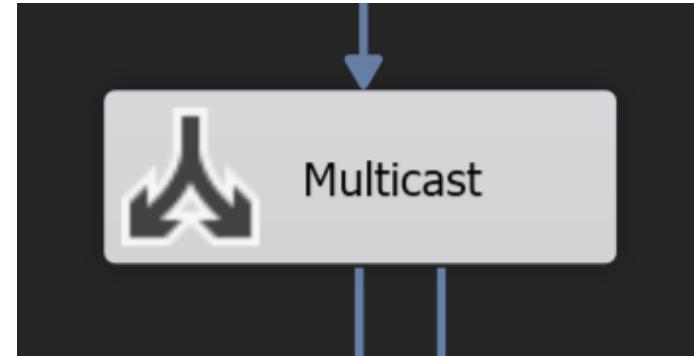
Merge Join Transform

- Perform INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN on sorted data
- Required Properties
 - Select Join Type
 - Select Columns to Output
 - Provide Output Column Alias



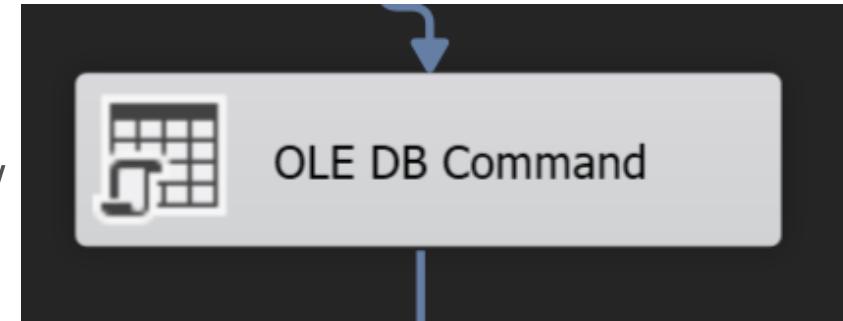
Multicast Transform

- create logical copies of input data as outputs
- Eg: one output to aggregate and summarize and the other output to retain detailed rows.



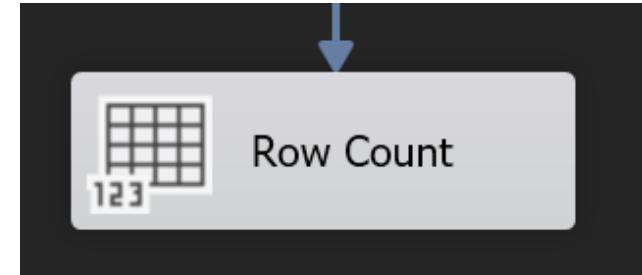
OLEDB Command Transform

- Perform a SQL Command for each input row
- Required Properties
 - Connection Manager
 - SQL Command to execute with Parameter Markers
 - Column Mapping to Parameters
- Can be used to inserts, updates, or deletes rows in a database table



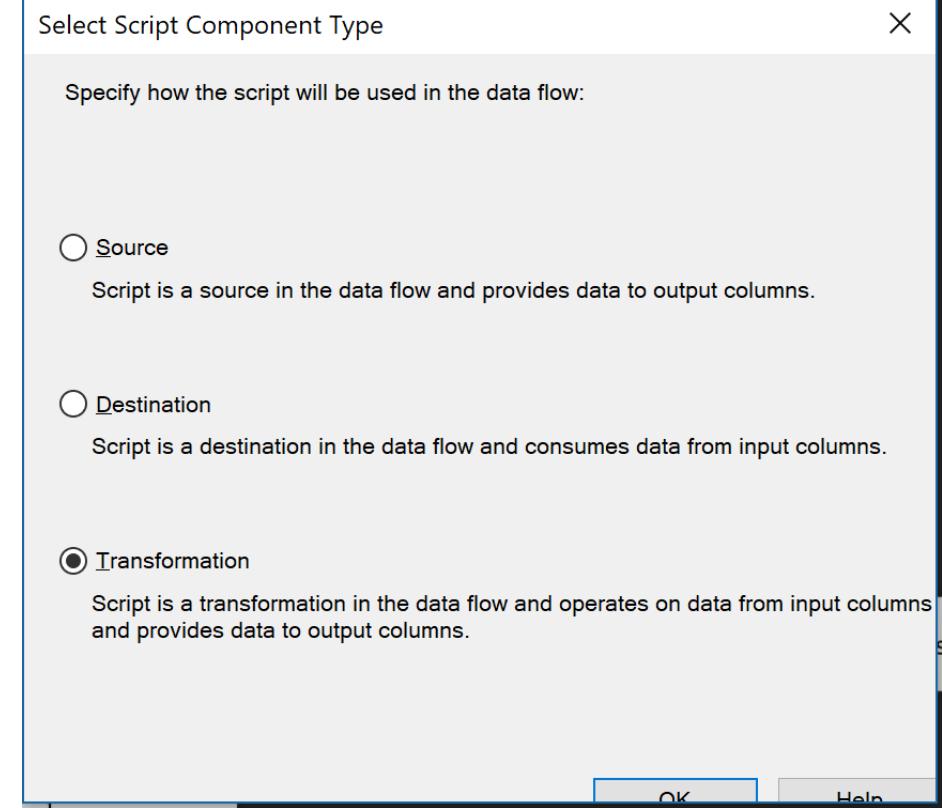
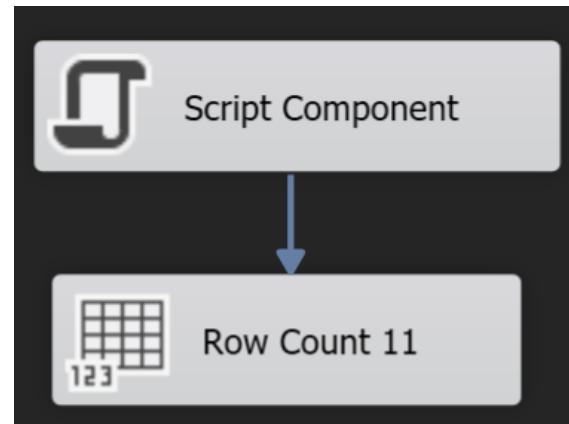
Row Count Transform

- Counts rows as they pass through a data flow and stores the final count in a variable
- Can be a transform or a destination
- Required Properties
 - VariableName



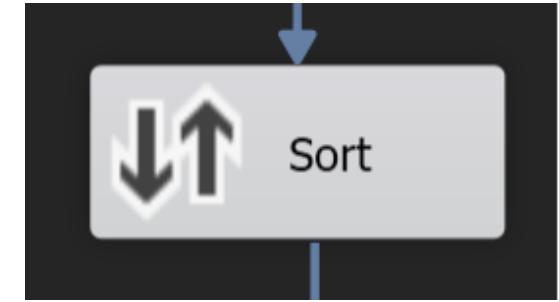
Script Component

- Use a script component as one of
 - Source
 - Transform
 - Destination



Sort Transform

- Sorts input data
- Is a blocking transform
 - All data must be sorted before data can proceed to next transform
- Avoid Sort transforms, if possible.
- Required Properties
 - Columns to sort on
 - Sorting Type (Ascending or Descending)
 - Sorting Order



Other Transforms

- Special use cases, such as
 - Change Data Capture
 - Data Quality Services

▲ Other Transforms	
	Audit
	CDC Splitter
	Character Map
	Copy Column
	Data Mining Query
	DQS Cleansing
	Export Column
	Fuzzy Grouping
	Fuzzy Lookup
	Import Column
	Percentage Sampling
	Pivot
	Row Sampling
	Term Extraction
	Term Lookup
	Unpivot

Data Flow Mastery

Lab



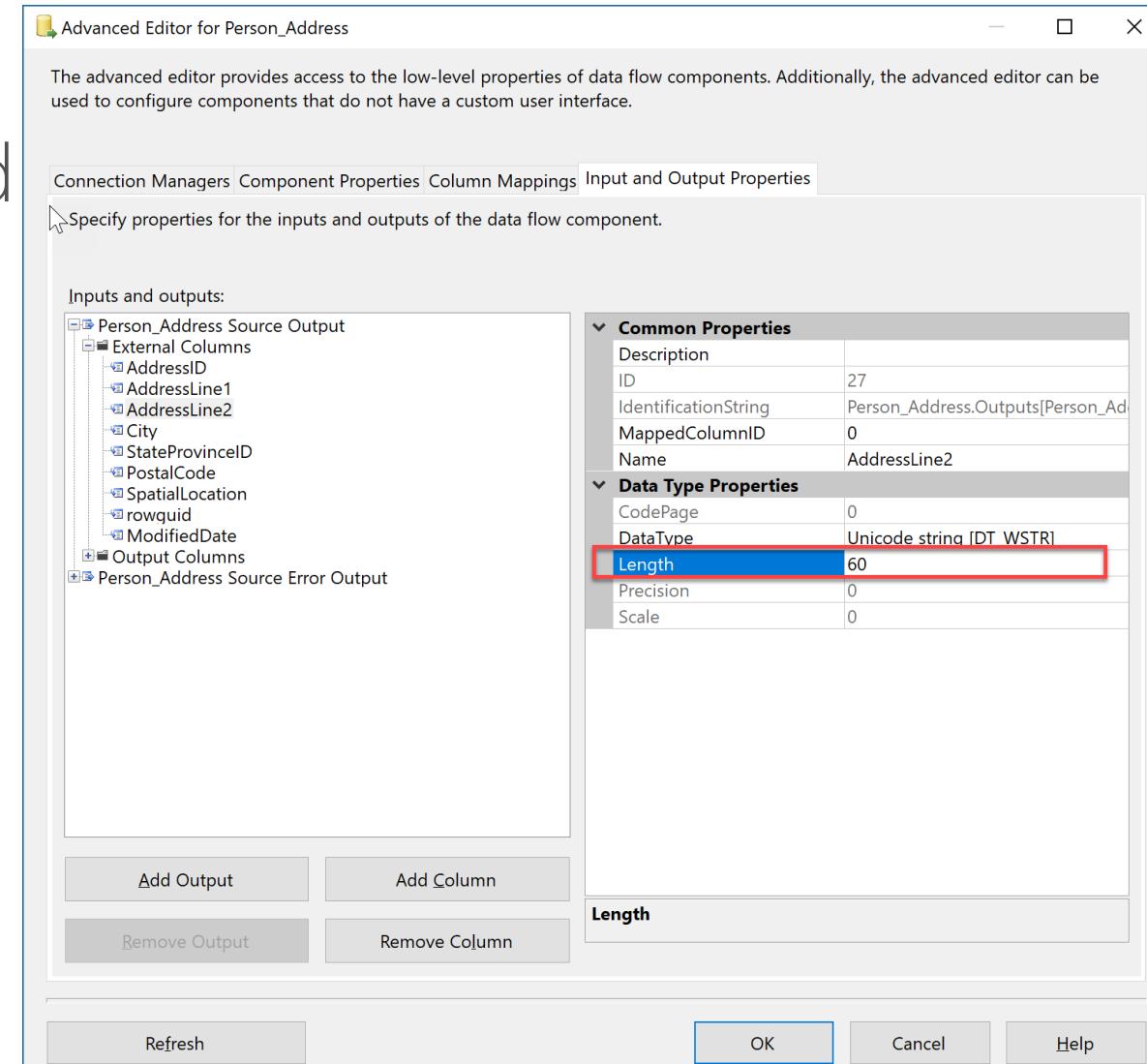
Module 4: Error Handling

Agenda

- Error Output
- Event Handler
- Transactions
- Checkpoints

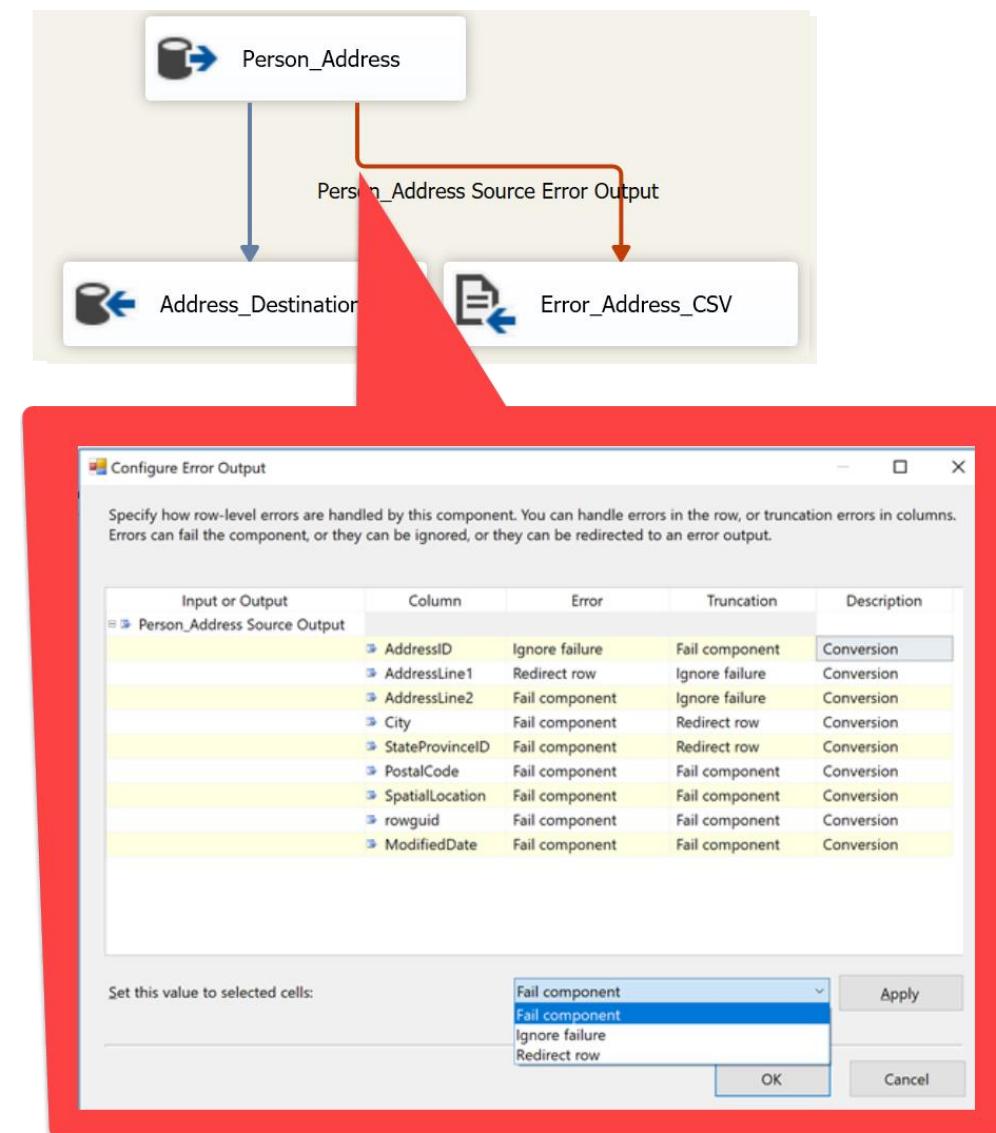
Types of Error within Data Flow

- Truncation – Data Source Column value is bigger SSIS Column data type and size defined in “Input and Output Properties” (via Show Advanced Editor)– leading to source value being truncated.
- Error – All other errors excluding Truncation Error



Error Outputs within Data Flow

- Determine how to handle errors for columns in Data Flow
 - Three Options
 1. Fail component (default setting) (e.g., in example, this setting will cause Data Flow Task to fail)
 2. Ignore Failure
 3. Redirect Row



Error Handling

Lab 1

Event Handler

- At run time, executables (packages and Foreach Loop, For Loop, Sequence, and task host containers) raise events.
 - e.g., OnError event is raised when an error occurs.
- Control flow canvas available for each event handler.
 - e.g., perform cleanup operation when error occurs or send email notifications.

Types of Event Handler

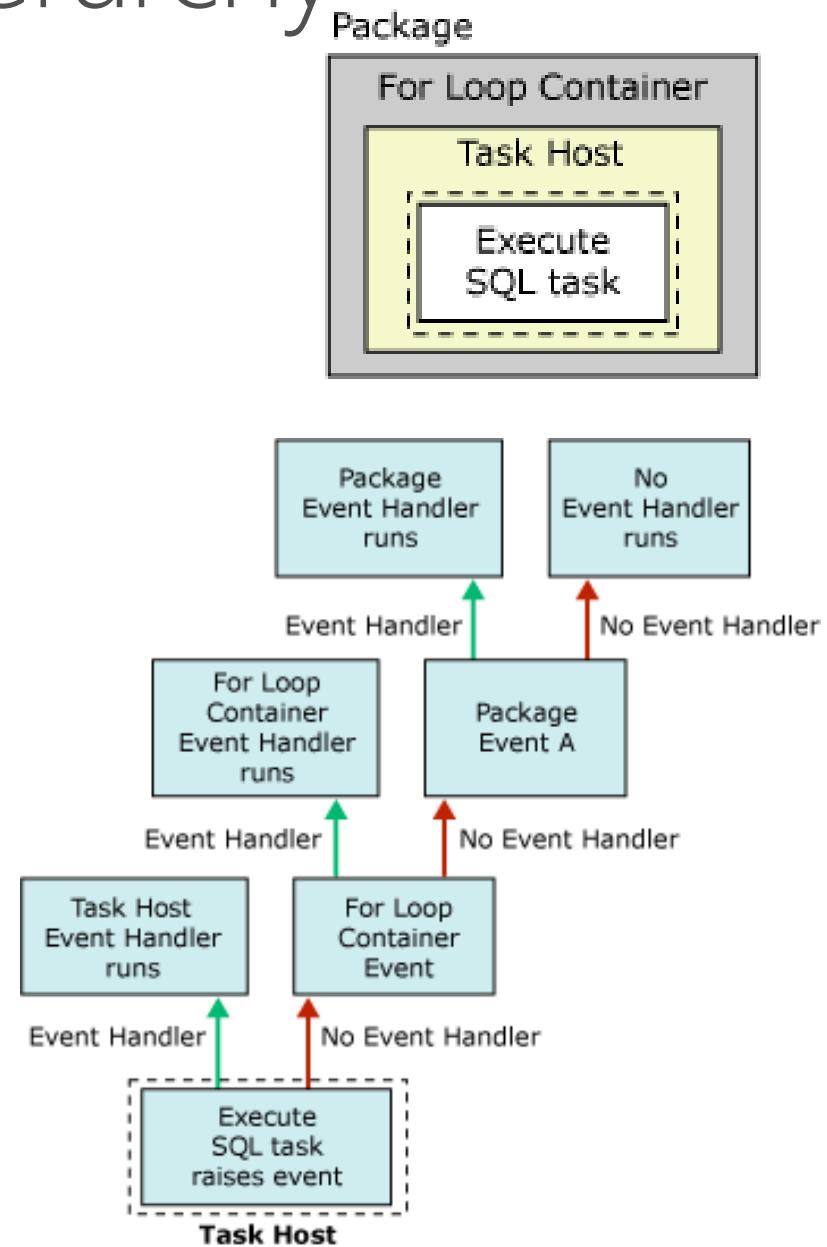
Event Handler	Event
OnPreExecute	This event is raised by an executable immediately before it runs.
OnPreValidate	This event is raised by an executable when its validation starts.
OnPostValidate	This event is raised by an executable when its validation is finished.
OnInformation	This event is raised during the validation and execution of an executable to report information. This event conveys information only, no errors or warnings.
OnWarning	This event is raised by an executable when a warning occurs.
OnError	This event is raised by an executable when an error occurs.
OnQueryCancel	This event is raised by an executable to determine whether it should stop running.
OnTaskFailed	This event is raised by a task when it fails.
OnExecStatusChanged	This event is raised by an executable when its execution status changes.
OnProgress	This event is raised by an executable when measurable progress is made by the executable.
OnVariableValueChanged	This event is raised by an executable when the value of a variable changes only if variable's RaiseChangedEvent property=True;
OnPostExecute	This event is raised by an executable immediately after it has finished running.

Event Handler and Container Hierarchy

- If an event has no event handler, the event is raised to the next container up the container hierarchy in a package.
- If this container has an event handler, the event handler runs in response to the event.
- If not, the event is raised to the next container up the container hierarchy.

Event Handler and Container Hierarchy (Example)

- In following example
 - Package has Execute SQL Task within For For Loop Container
 - Package has OnError Event handler defined
- Notice: how Events are bubbled up until an Event Handler is found.



Event Handler

Lab 2

Transactions

- Execute tasks as atomic units
- Transaction can be used to rollback when error occurs
- Requires distributed transaction coordinator
- All container types—packages, the For Loop, Foreach Loop, and Sequence containers, and the task hosts that encapsulate each task—can be configured to use transactions.

TransactionOption	Supported
NotSupported	
Supported	
Required	

TransactionOption Property	Description
Required	Container starts a transaction or joins its parent containers existing transaction.
Supported	Container does not start a transaction, but joins any transaction started by its parent container.
NotSupported	Container does not start a transaction or join an existing transaction

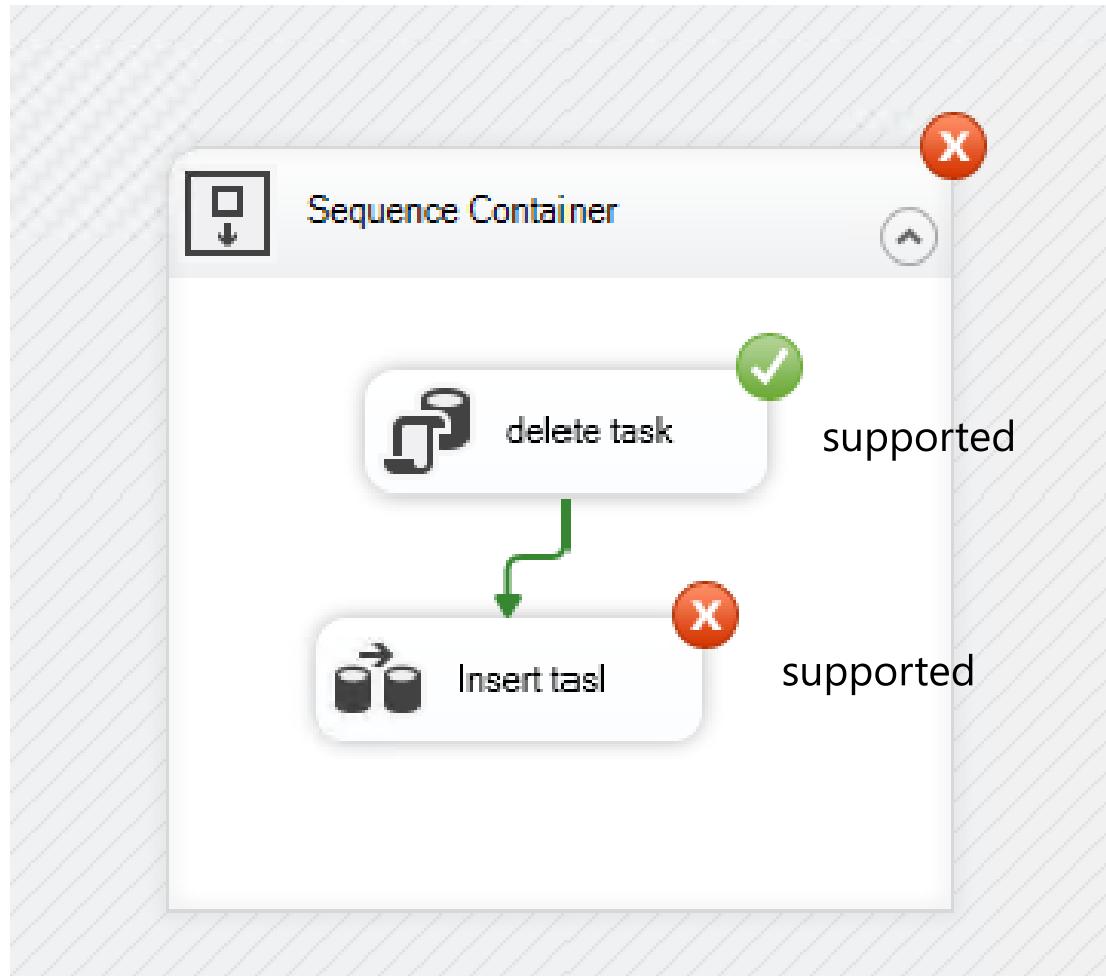
Inherited Transactions

- Child package can inherit the parent package transaction if following are met:
 - Parent Package TransactionOption=Required (or Supported and has joined transaction)
 - Child package is invoked by an Execute Package task
 - Execute Package task's TransactionOption=Supported

	<p>時序容器</p> <p>required</p>	No
	<p>時序容器</p> <p>supported</p>	Yes
	<p>時序容器</p> <p>supported</p>	not supported
Yes	<p>時序容器</p> <p>supported</p>	<p>supported</p> <p>required</p> <p>required</p>

❑ Transactions

IsolationLevel	Serializable
TransactionOption	Supported



100 %

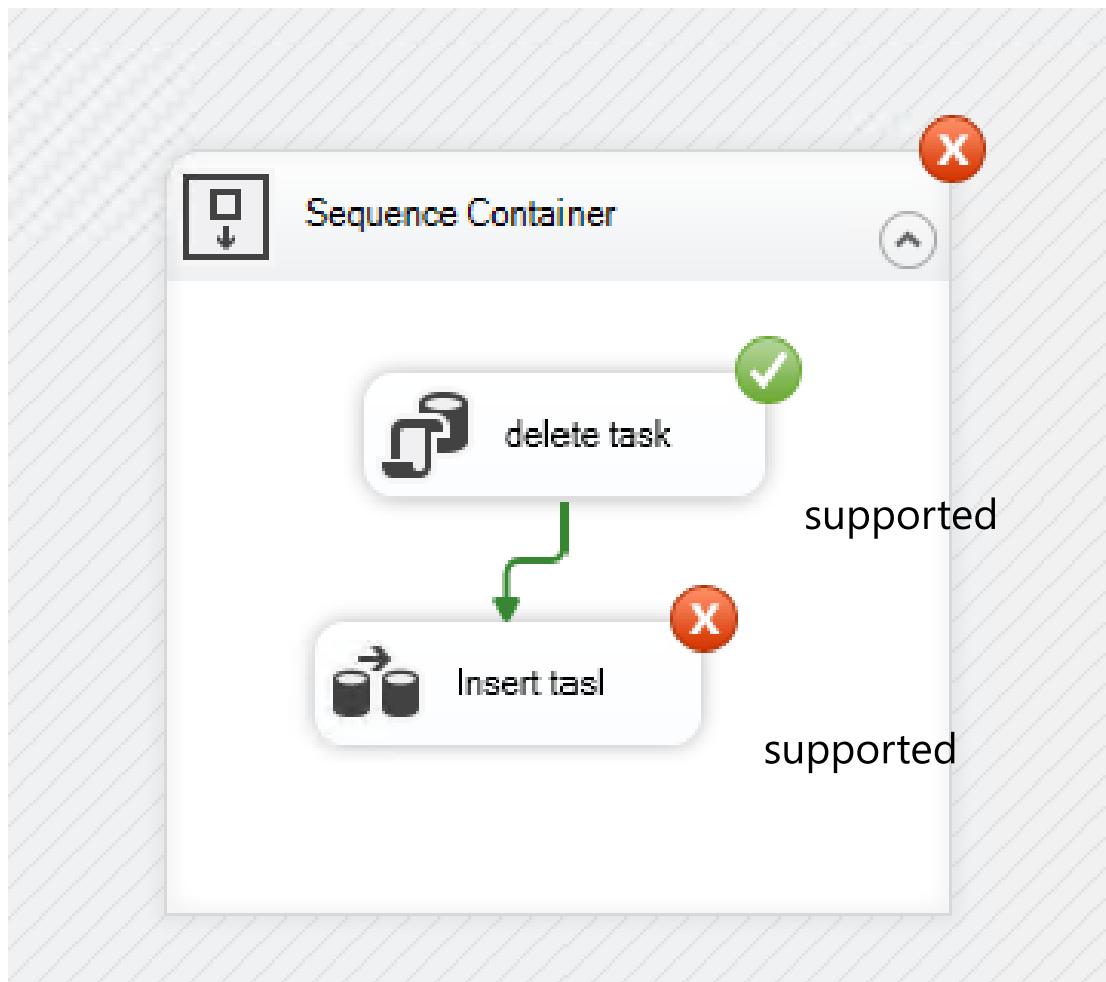
Results

id
1
2

Results

id
1

Transactions	
IsolationLevel	Serializable
TransactionOption	Required



100 %

Results

	id
1	1
2	2

100 %

Results

	id
1	1
2	2



Transaction Nesting

- Inner transactions are not rolled back as result of outer transaction failure. Careful, might care unintended behavior.
- Nested transaction scan be creating by adjusting the TransactionOption property.
- For example:
 - Package Property set to Required.
 - Container Property set to Not Supported.
 - Task Property set to Supported.

Restart Packages using Checkpoints

- Restart failed packages from the point of failure, instead of rerunning the whole package.
- Package execution is written to a checkpoint file.
- When the failed package is rerun, the checkpoint file is used to restart the package from the point of failure.
- If the package runs successfully, the checkpoint file is deleted and recreated on next run.
- Checkpoints are not saved for data flow task.
- Checkpoints are not saved for container items (For Loop & Foreach Loop).

Implementing CheckPoints

Package Property	Description
CheckpointFileName	<p>Contain information about the package state, including the current values of variables.</p> <ul style="list-style-type: none">• Package Level protection does not protect checkpoint files.• Can contain sensitive values.• Use an operating system access control list (ACL) to secure the location or folder where you store the file• Suggestion: @[\$Project::CheckpointPath] +@[System::PackageName] + "_SSIS_.CHK" or @[\$Project::CheckpointPath] +@[System::PackageName] + "_SSIS_Checkpoint.xml"
CheckpointUsage = IfExists	IfExists means If the checkpoint file exists, the package restarts from the point of the previous execution failure; otherwise, it runs from the start of the package workflow.
SaveCheckpoints = True	Indicates whether the package saves checkpoints. This property must be set to True to restart a package from a point of failure.
Task Property	Description
FailPackageOnFailure = True	Indicate package fails when executable task fails. Also used to identify restart points if checkpoints are used. Must be set to use checkpoints.
FailParentOnFailure = False	This value should be set to true for containers, such as a sequence.



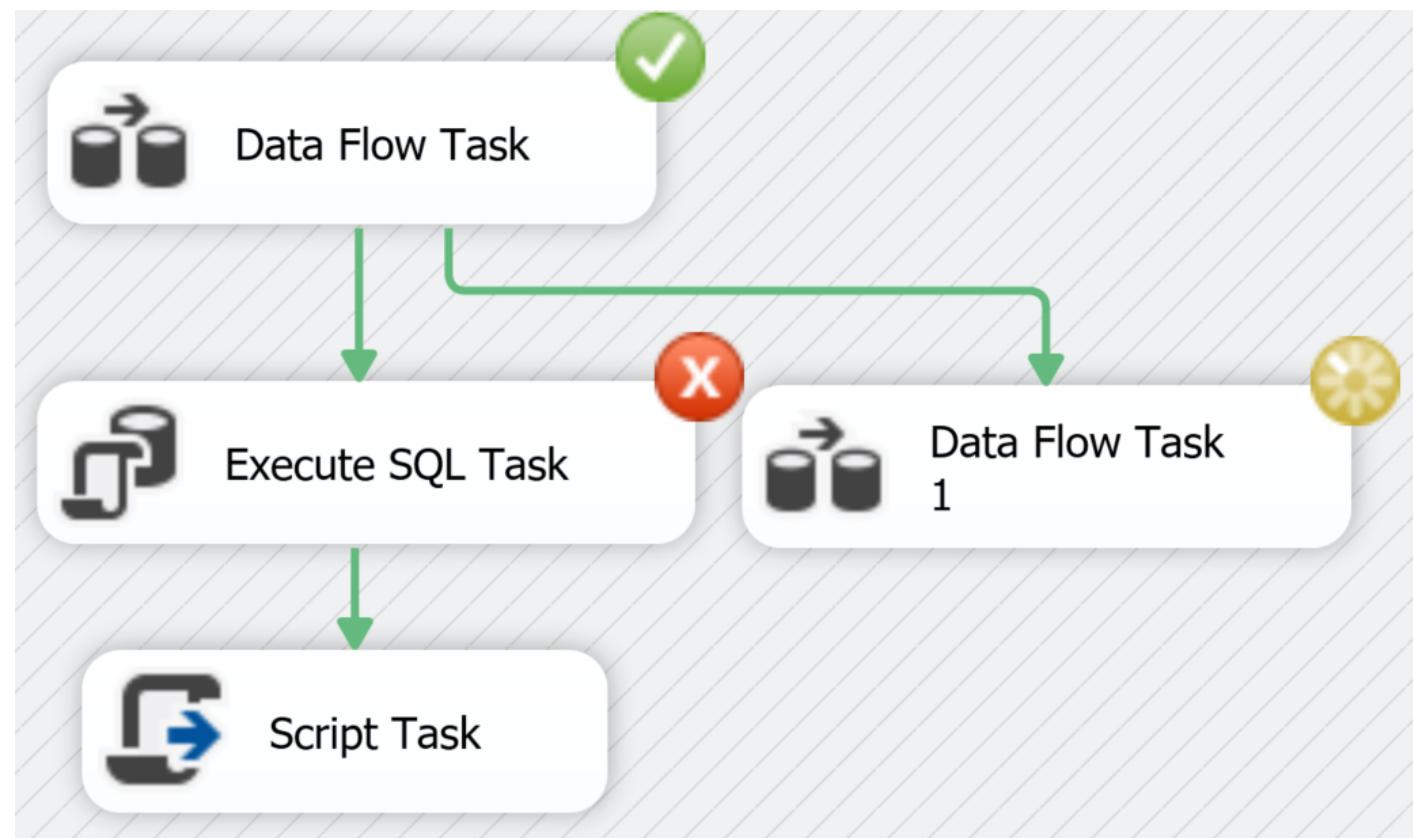
Module 6: Troubleshooting

Agenda

- SSDT Features
 - Progress Reporting
 - Data Flow - Data Viewer
 - Break Points
 - Watch List
- Integration Services Dashboard

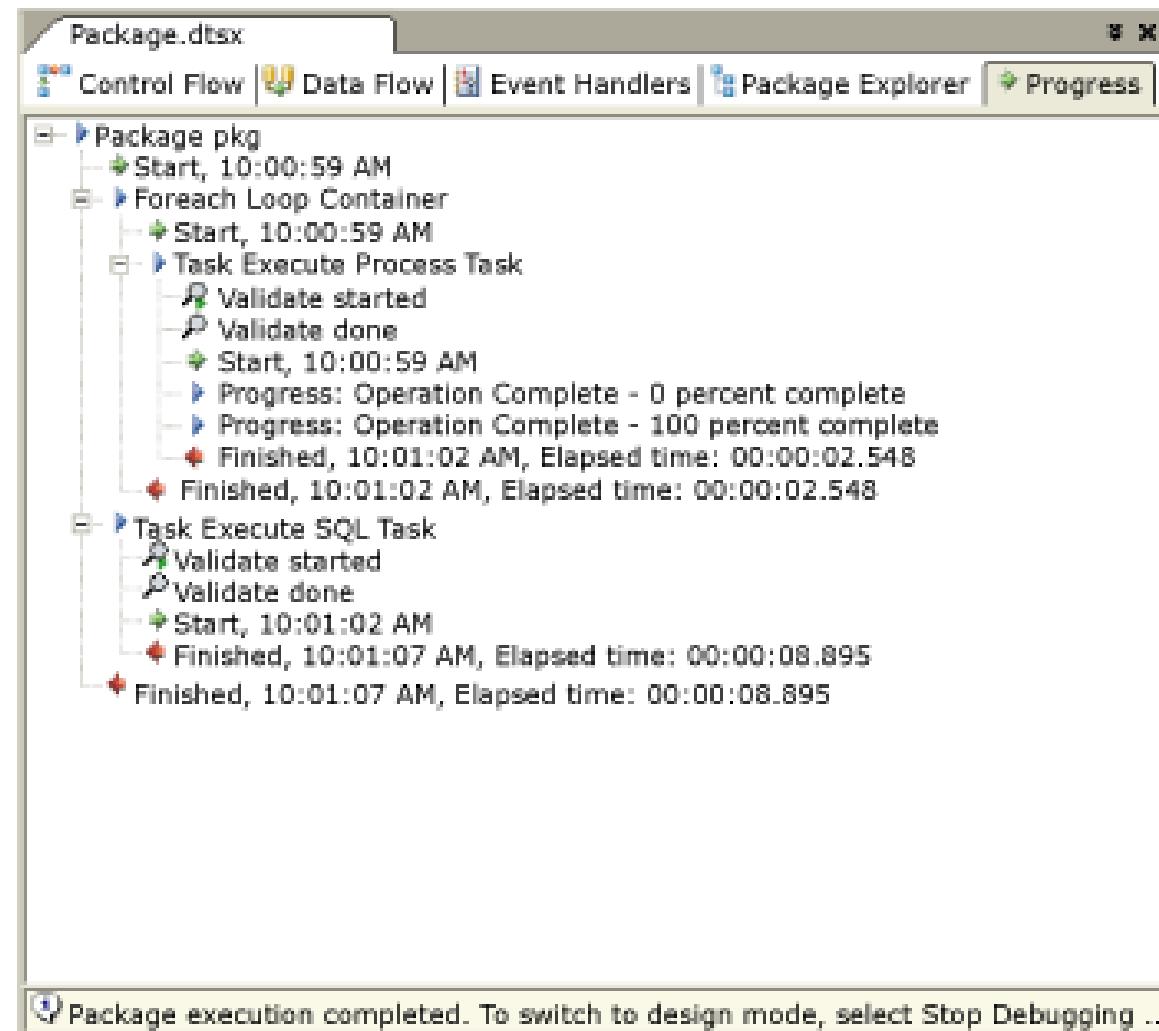
Progress Reporting – task/component states

- Visible only when executing the package in SSDT
- Progress States are:
 - Running -
 - Successfully completed –
 - Failed –
 - No color – waiting for execution



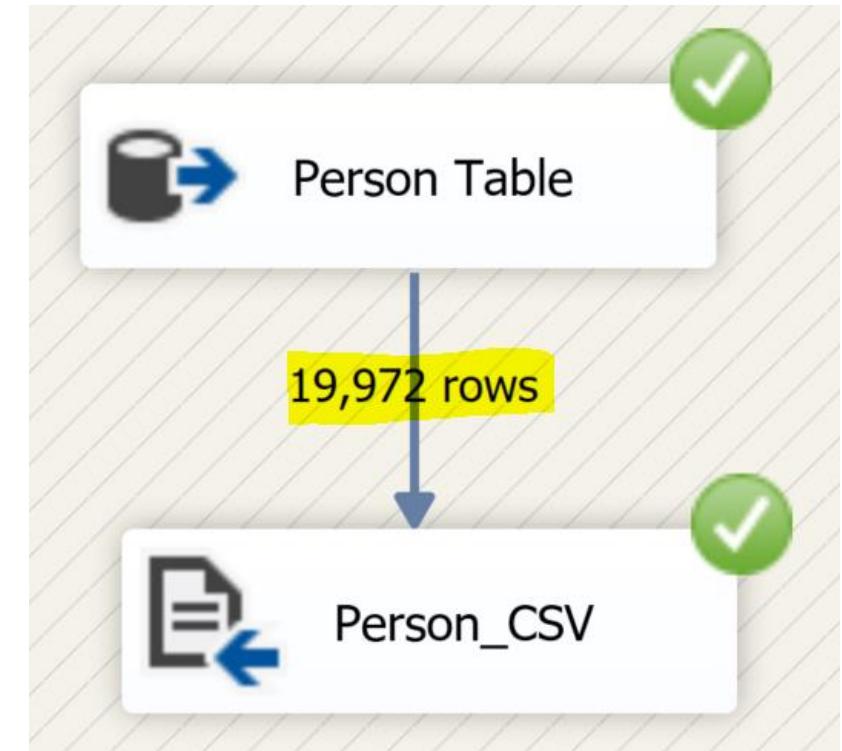
Progress Reporting - messages

- Displays informational, warning, and error messages in Progress Tab of SSDT
- Percentage completion
- Progress of the various phases
 - Pre-Execute
 - Validate
 - Execution



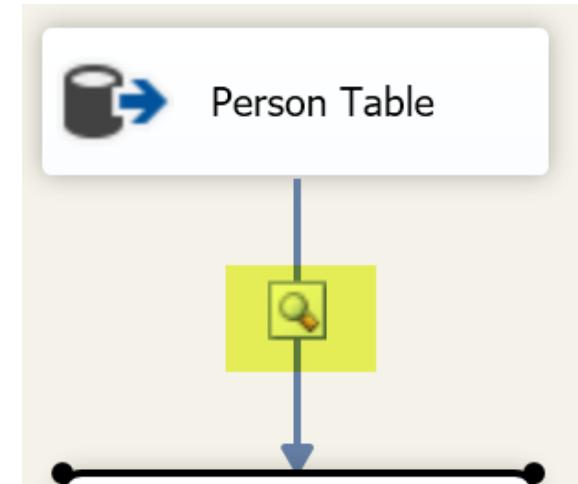
Progress Reporting – Data Flow Row Count

- Number of rows that have passed through a path is displayed and is updated progressively
- Row counts are visible only when executing the package in SSDT



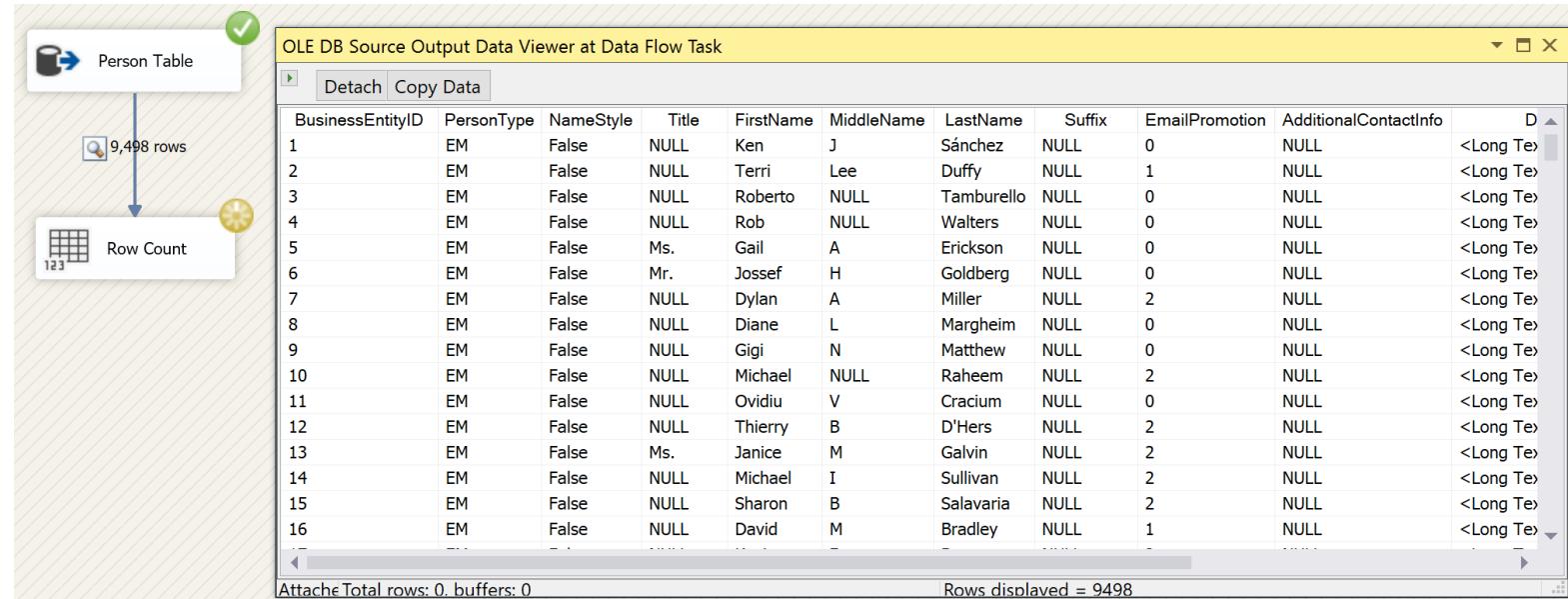
Data Flow - Data Viewer

- SSIS Designer feature
- Display data between two components in a data flow
- Example of uses
 - display extracted data from a data source
 - display before and after a transformation
 - display before the data is loaded into destination
- Data viewer results are viewable only when executing the package in SSDT



Data Flow - Data Viewer (2)

- Displays one buffer at a time.
- Play button to proceed to next buffer
- Detach – stop data viewer
- Attach – start data viewer
- Can detach & attach multiple times while package is running



Breakpoints

- Set breakpoints by enabling break conditions
- Some tasks and containers include special break conditions for setting breakpoints
 - Eg: can break at start of each For Loop container iteration.



Set Breakpoints - Execute SQL Task

Select the breakpoints in the task, For Loop, Foreach Loop, or Sequence to enable. Optionally, select the number of times a breakpoint is ignored before execution is suspended on the breakpoint.

Enabled	Break Condition	Hit Count Type	Hit Count
<input checked="" type="checkbox"/>	Break when the container receives the OnPreExecute event	Always	0
<input type="checkbox"/>	Break when the container receives the OnPostExecute event	Always	0
<input type="checkbox"/>	Break when the container receives the OnError event	Hit count equals	0
<input type="checkbox"/>	Break when the container receives the OnWarning event	Hit count greater than or equal to	0
<input type="checkbox"/>	Break when the container receives the OnInformation event	Hit count multiple	0
<input type="checkbox"/>	Break when the container receives the OnTaskFailed event	Always	0
<input type="checkbox"/>	Break when the container receives the OnProgress event	Always	0
<input type="checkbox"/>	Break when the container receives the OnQueryCancel event	Always	0
<input type="checkbox"/>	Break when the container receives the OnVariableValueChanged event	Always	0
<input type="checkbox"/>	Break when the container receives the OnCustomEvent event	Always	0

OK Cancel Help

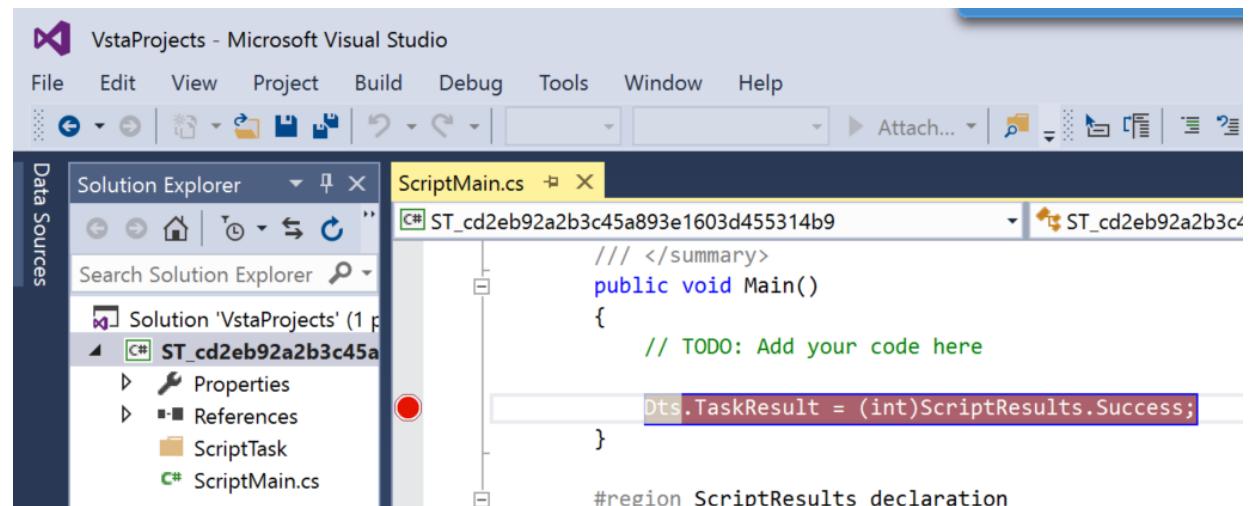
Breakpoints – Hit count

- Use Hit Count to modify breakpoint behavior

Hit count type	Description
Always	Execution is always suspended when the breakpoint is hit.
Hit count equals	Execution is suspended when the number of times the breakpoint has occurred is equal to the hit count.
Hit count greater than or equal to	Execution is suspended when the number of times the breakpoint has occurred is equal to or greater than the hit count.
Hit count multiple	Execution is suspended when a multiple of the hit count occurs. For example, if you set this option to 5, execution is suspended every fifth time.

Breakpoints – script task or script component

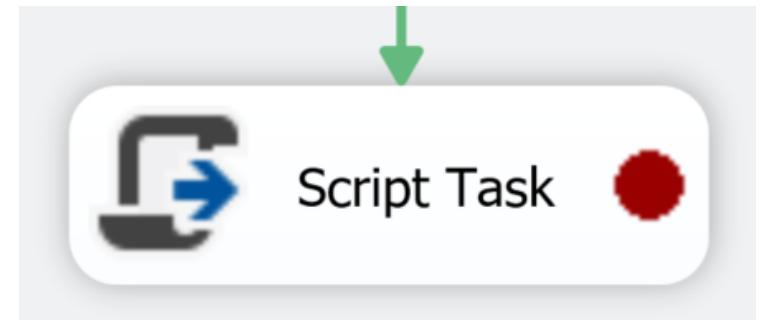
- You can set breakpoints within Microsoft Visual Studio Tools for Applications (VSTA).
- can manage the breakpoints in VSTA or also using the Set Breakpoints dialog box that SSIS Designer provides



The screenshot shows the Microsoft Visual Studio interface with the title bar "VstaProjects - Microsoft Visual Studio". The menu bar includes File, Edit, View, Project, Build, Debug, Tools, Window, and Help. The toolbar has various icons for file operations. The Solution Explorer on the left shows a solution named "VstaProjects" with one project "ST_cd2eb92a2b3c45a" containing "Properties", "References", "ScriptTask", and "ScriptMain.cs". The "ScriptMain.cs" file is open in the code editor, showing the following code:

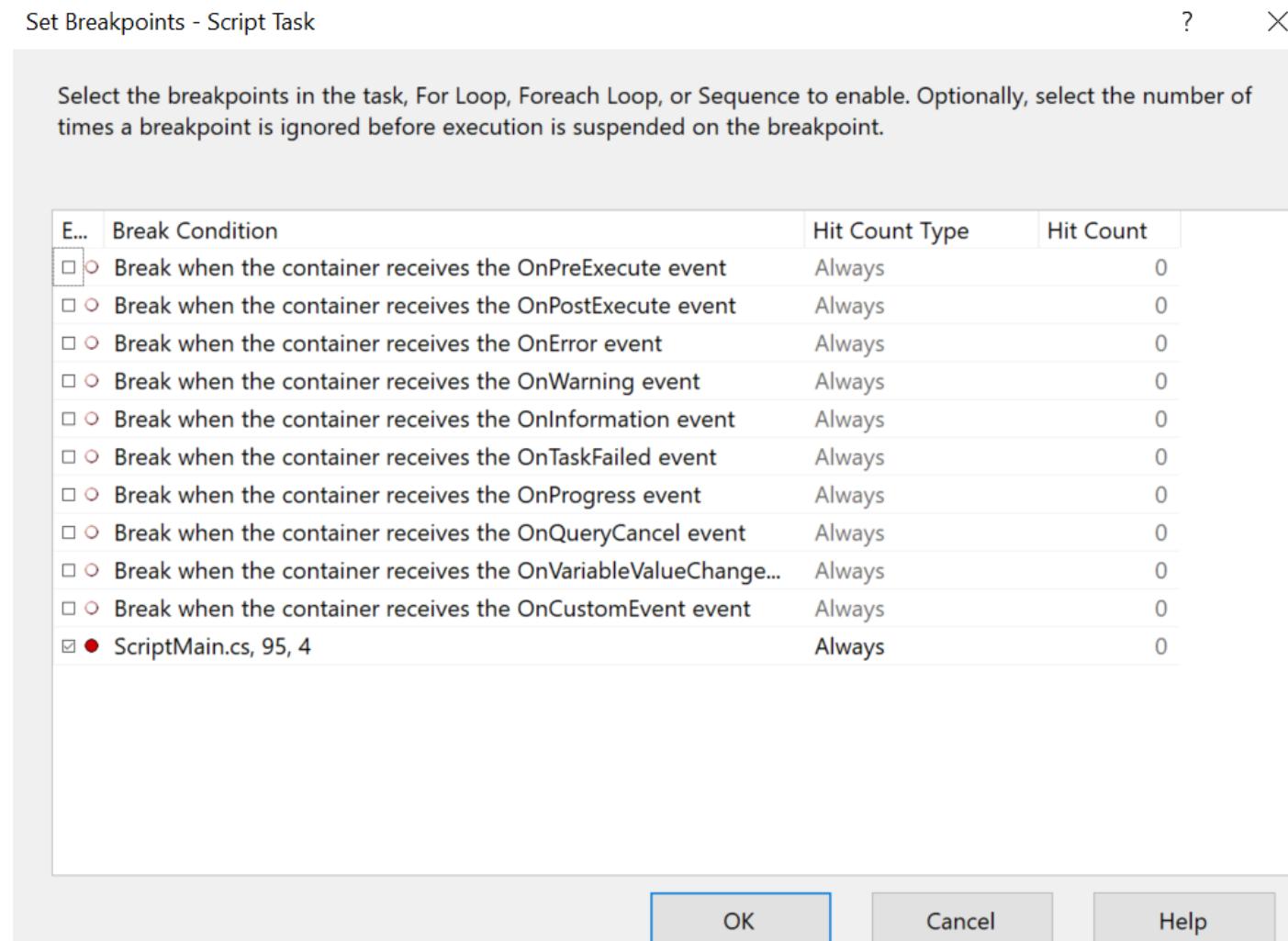
```
//</summary>
public void Main()
{
    // TODO: Add your code here
    Dts.TaskResult = (int)ScriptResults.Success;
}
```

A red circular breakpoint icon is visible on the line "Dts.TaskResult = (int)ScriptResults.Success;".



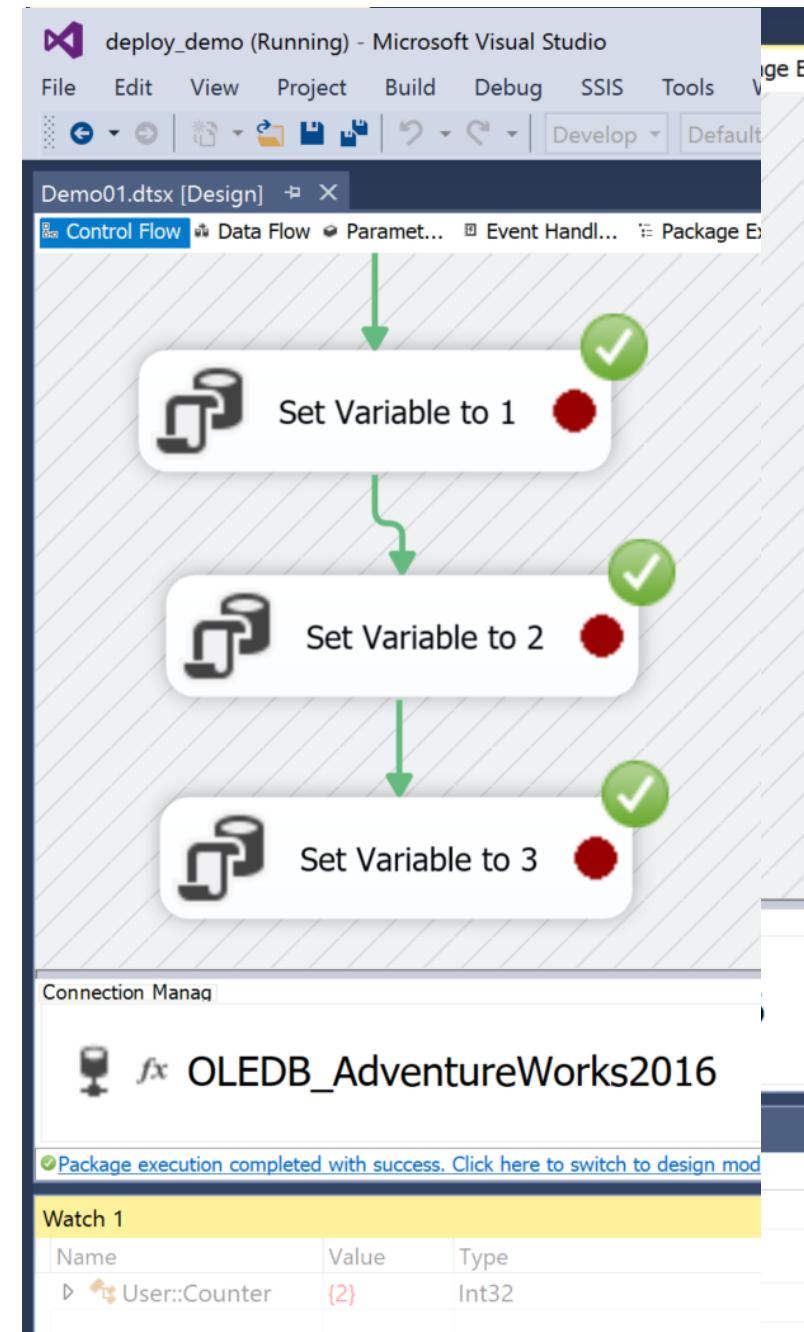
Breakpoints – script task or script component (2)

- Set Breakpoints dialog box also includes the script breakpoints
 - Appear at bottom breakpoint list
 - display line number and function name where breakpoint appears



Watch Window

- Add SSIS variables (Scope::VariableName) to Watch List when at break to view the variable value
- You can set breakpoints and observe variable values before executing the task



Integration Services Dashboard

- View current state of packages
 - Running
 - Succeeded
 - Failed
 - View Error Message for failures
- View Parameter values used
- Property overrides
- Which Environment was used
- Duration for each control flow task

Integration Services Dashboard

- Running State

Overview - 2018-0...P-N48JLH3\SQL2017 -> X

SQL Server

Overview

on DESKTOP-N48JLH3\SQL2017 at 2018-05-23 12:58:08 AM

This report provides an overview of the package tasks and parameters, including execution or validation information.

[View Messages](#)

[View Performance](#)

Execution Information

Operation ID	310060
Package	Demo\deploy_demo\Demo01.dtsx
Environment	.\\Dev
Status	Running
Machine	DESKTOP-N48JLH3

Duration (sec)	25.632
Start Time	2018-05-23 12:57:42 AM
End Time	
Caller	NORTHAMERICA\\jaahmed

Parameters Used

Name	Value	Data Type
CALLER_INFO		String
CheckPointFilePath	c:\\temp\\	String
DUMP_EVENT_CODE	0	String
DUMP_ON_ERROR	False	Boolean
DUMP_ON_EVENT	False	Boolean
LOGGING_LEVEL	1	Int32
OLEDB_AdventureWorks2016_ServerName	MOGUPTA-PC01	String
OLEDB_AdventureWorksDW2016_ServerName	jaahmed	String
SYNCHRONIZED	False	Boolean

Integration Services Dashboard

- Succeeded State

Overview - 2018-0...P-N48JLH3\SQL2017

This report provides an overview of the package tasks and parameters, including execution or validation information.

[View Messages](#)

[View Performance](#)

Execution Information

Operation ID	310060
Package	Demo\deploy_demo\Demo01.dtsx
Environment	.\\Dev
Status	Succeeded
Machine	DESKTOP-N48JLH3

Execution Overview

Filter: Result: All; (3 more)

Result	Duration (sec)	Package Name	Task Name	Execution Path
Succeeded	25.032	Demo01.dtsx	Demo01	\Demo01
Succeeded	24.562	Demo01.dtsx	Data Flow Task	\Demo01\Data Flow Task
Succeeded	0.157	Demo01.dtsx	Set Variable to 1	\Demo01\Set Variable to 1
Succeeded	0.031	Demo01.dtsx	Set Variable to 2	\Demo01\Set Variable to 2
Succeeded	0.032	Demo01.dtsx	Set Variable to 3	\Demo01\Set Variable to 3

Parameters Used

Name	Value	Data Type
CALLER_INFO		String
CheckPointFilePath	c:\\temp\\	String
DUMP_EVENT_CODE	0	String
DUMP_ON_ERROR	False	Boolean
DUMP_ON_EVENT	False	Boolean
LOGGING_LEVEL	1	Int32
OLEDB_AdventureWorks2016_ServerName	MOGUPTA-PC01	String
OLEDB_AdventureWorksDW2016_ServerName	jaahmed	String
SYNCHRONIZED	False	Boolean

Property Overrides

Property Path	Property Value
---------------	----------------

Integration Services Dashboard

- View messages similar to the Execution Results Tab in SSDT

Message Type	Message Time	Message	Message Source Name	Subcomponent Name	Execution Path
OnPostExecute	2018-05-23 12:58:10 AM	Demo01:Finished, 12:58:10 AM, Elapsed time: 00:00:25.032.	Demo01		\Demo01
OnPostExecute	2018-05-23 12:58:10 AM	Set Variable to 3:Finished, 12:58:10 AM, Elapsed time: 00:00:00.032.	Set Variable to 3		\Demo01\Set Variable
OnPostValidate	2018-05-23 12:58:10 AM	Set Variable to 3:Validation is complete.	Set Variable to 3		\Demo01\Set Variable
OnPreValidate	2018-05-23 12:58:10 AM	Set Variable to 3:Validation has started.	Set Variable to 3		\Demo01\Set Variable
OnPreExecute	2018-05-23 12:58:10 AM	Set Variable to 3:Start, 12:58:10 AM.	Set Variable to 3		\Demo01\Set Variable
OnPostExecute	2018-05-23 12:58:10 AM	Set Variable to 2:Finished, 12:58:10 AM, Elapsed time: 00:00:00.031.	Set Variable to 2		\Demo01\Set Variable
OnPostValidate	2018-05-23 12:58:10 AM	Set Variable to 2:Validation is complete.	Set Variable to 2		\Demo01\Set Variable
OnPreValidate	2018-05-23 12:58:10 AM	Set Variable to 2:Validation has started.	Set Variable to 2		\Demo01\Set Variable
OnPreExecute	2018-05-23 12:58:10 AM	Set Variable to 2:Start, 12:58:10 AM.	Set Variable to 2		\Demo01\Set Variable
OnPostExecute	2018-05-23 12:58:10 AM	Set Variable to 1:Finished, 12:58:10 AM, Elapsed time: 00:00:00.157.	Set Variable to 1		\Demo01\Set Variable
OnPostValidate	2018-05-23 12:58:10 AM	Set Variable to 1:Validation is complete.	Set Variable to 1		\Demo01\Set Variable
OnPreValidate	2018-05-23 12:58:10 AM	Set Variable to 1:Validation has started.	Set Variable to 1		\Demo01\Set Variable
OnPreExecute	2018-05-23 12:58:10 AM	Set Variable to 1:Start, 12:58:10 AM.	Set Variable to 1		\Demo01\Set Variable
OnPostExecute	2018-05-23 12:58:10 AM	Data Flow Task:Finished, 12:58:10 AM, Elapsed time: 00:00:24.562.	Data Flow Task		\Demo01\Data Flow T
OnInformation	2018-05-23 12:58:10 AM	Data Flow Task:Information: Cleanup phase is beginning.	Data Flow Task	SSIS.Pipeline	\Demo01\Data Flow T
OnInformation	2018-05-23 12:58:10 AM	Data Flow Task:Information: "Person_CSV" wrote 19972 rows.	Data Flow Task	SSIS.Pipeline	\Demo01\Data Flow T
OnInformation	2018-05-23 12:58:10 AM	Data Flow Task:Information: The processing of file "c:\temp\person.csv" has ended.	Data Flow Task	Person_CSV [56]	\Demo01\Data Flow T
OnInformation	2018-05-23 12:58:10 AM	Data Flow Task:Information: Post Execute phase is	Data Flow Task	SSIS.Pipeline	\Demo01\Data Flow T

SSIS DB Queries

- Quick insight on
 - Running packages,
 - Performance History
 - Run outcome
 - DataFlow statistics and performances
 - Lookup component memory usage
 - Lookup component duplicate warnings
 - Errors
 - Memory Warnings
- Provided queries also used in [ssis-dashboard](#) project.
- <https://github.com/yorek/ssis-queries>

ssis-dashboard project

- web-based SQL Server Integration Services Dashboard and a set of REST API to monitor execution of SQL Server Integration Services Packages.

SQL Server Integration Services Monitoring Dashboard v 0.5.2 (beta)

DEMO @2014-08-21 20:27:56.895000

37 Executed View	0 Running Packages View	0 Cancelled View	16 Failed View	1 Halted View	20 Succeeded View
--	---	--	--------------------------------------	-------------------------------------	---

All Packages

#	Project	Package	Status	Start Time	Stop Time	Elapsed (Min)	Warnings	Errors
82	Another Test (v.9)	Master.dtsx	Succeeded	2014-08-20 15:52:48	2014-08-20 15:53:22	0.566666	20	None
81	Another Test (v.9)	Master.dtsx	Ended Unexpectedly	2014-08-20 15:51:35	2014-08-20 15:51:38	0.050000	4	None
79	Test Project 2 (v.8)	Master.dtsx	Succeeded	2014-08-20 15:45:30	2014-08-20 15:46:03	0.550000	20	None
78	Test Project 2 (v.8)	Master.dtsx	Succeeded	2014-08-20 15:44:45	2014-08-20 15:45:19	0.566666	20	None
77	Test Project 2 (v.8)	Master.dtsx	Succeeded	2014-08-20 15:44:02	2014-08-20 15:44:35	0.550000	20	None

Showing 1 to 5 of 15 entries

Previous [1](#) [2](#) [3](#) Next

Executables

5 records per page

#	Generator	Package	Source	Start Time	Stop Time	Elapsed (Min)
---	-----------	---------	--------	------------	-----------	---------------



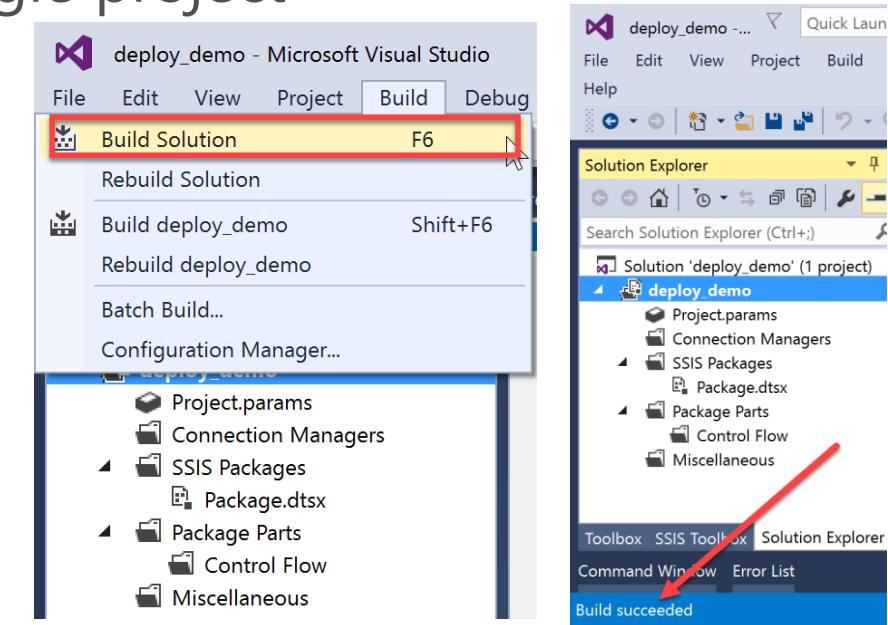
Module 7: Deployment

Agenda

- How to deploy a package using SSDT?
- How to use ISPAC to deploy an package?
- How to setup Environments in SSIS CatalogDB?

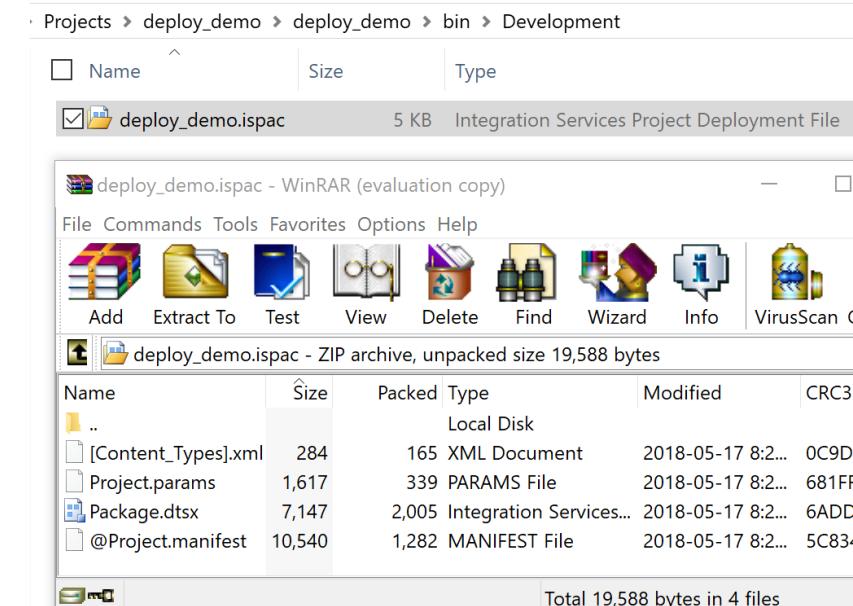
Building Project

- Bundles all packages, project parameter, shared connections to an .ispac file in \bin\Development\ subfolder
- Two options to build
 - Build Solution – builds all projects within a solution (F6)
 - i.e. a solution can have multiple SSIS projects
 - Build <Project_Name> - build only the currently single project (Shift+F6)



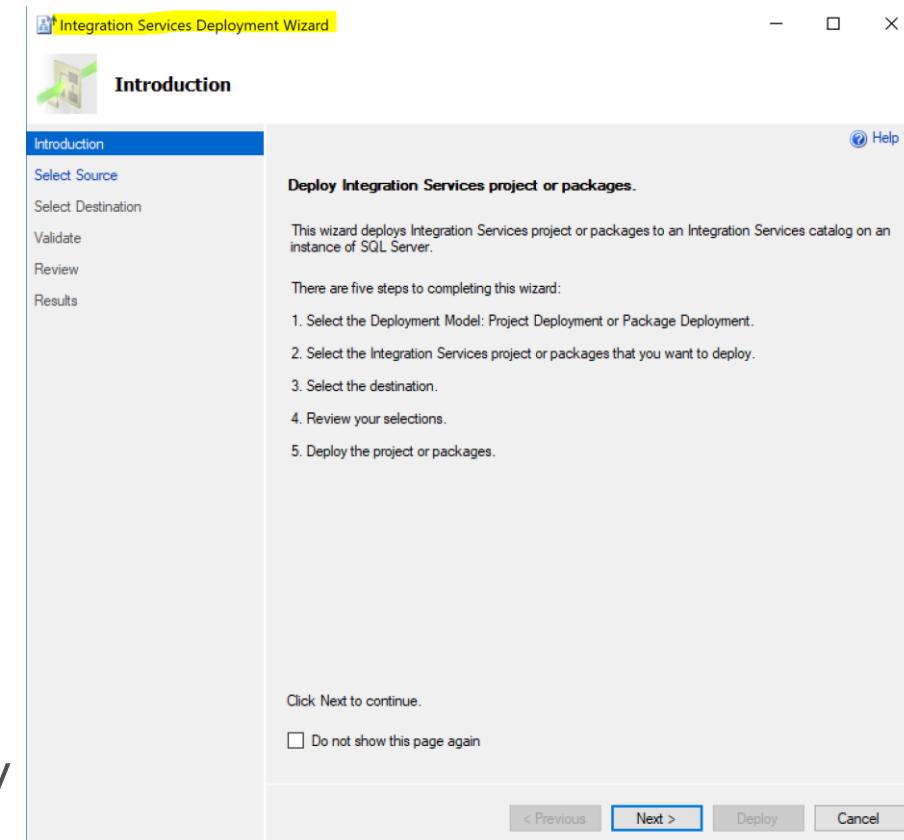
.ispac file

- Output of a solution or project Build process
 - It is a zip file containing all packages, project parameter, shared connections
 - You can recreate Visual Studio Solution if you have .ispac file using “Integration Services Import Project Wizard” from SSDT
 - Select Project Deployment File (i.e. .ispac file)
- Or
- Select Project from SSISDB Catalog



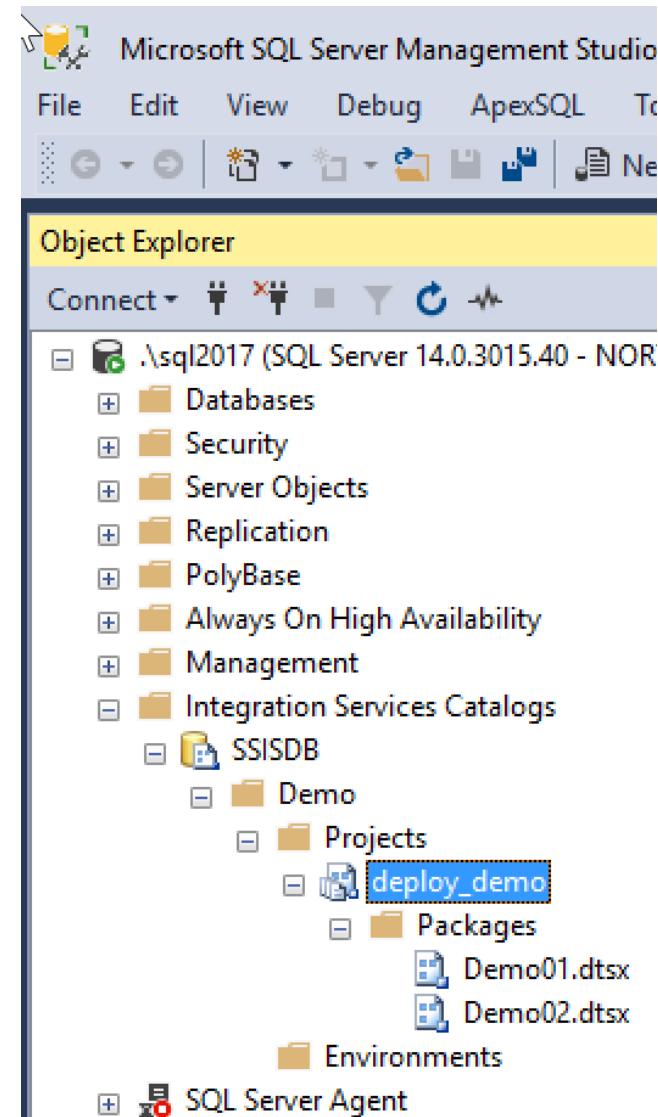
Deploying .ispac

- Deployed with Integration Services Deployment Wizard
- Three ways to call it
 1. By itself as a wizard, from Windows Start bar
 1. Need to specify .ispac file and deploy location
 2. Within SSDT
 - Right-click project in Solution Explorer and select Deploy
 - Filename is automatically passed as argument to wizard
 - Need to specify deploy location on first deploy
 - Deploy Location saved for subsequent deploys in Project Properties\Configuration Properties\Deployment
 3. Double-clicking on .ispac file in windows explorer
 1. Filename is automatically passed as argument to wizard
 2. Need to specify Deploy location
- DBAs do not require SSDT to deploy



Deployment Location

- SSISDB Catalog is the preferred deployment location
- Use SSMS to browse, execute deployed packages



How to deploy using SSDT?

Demo

How to deploy using ISPAC file?

Demo

Environment

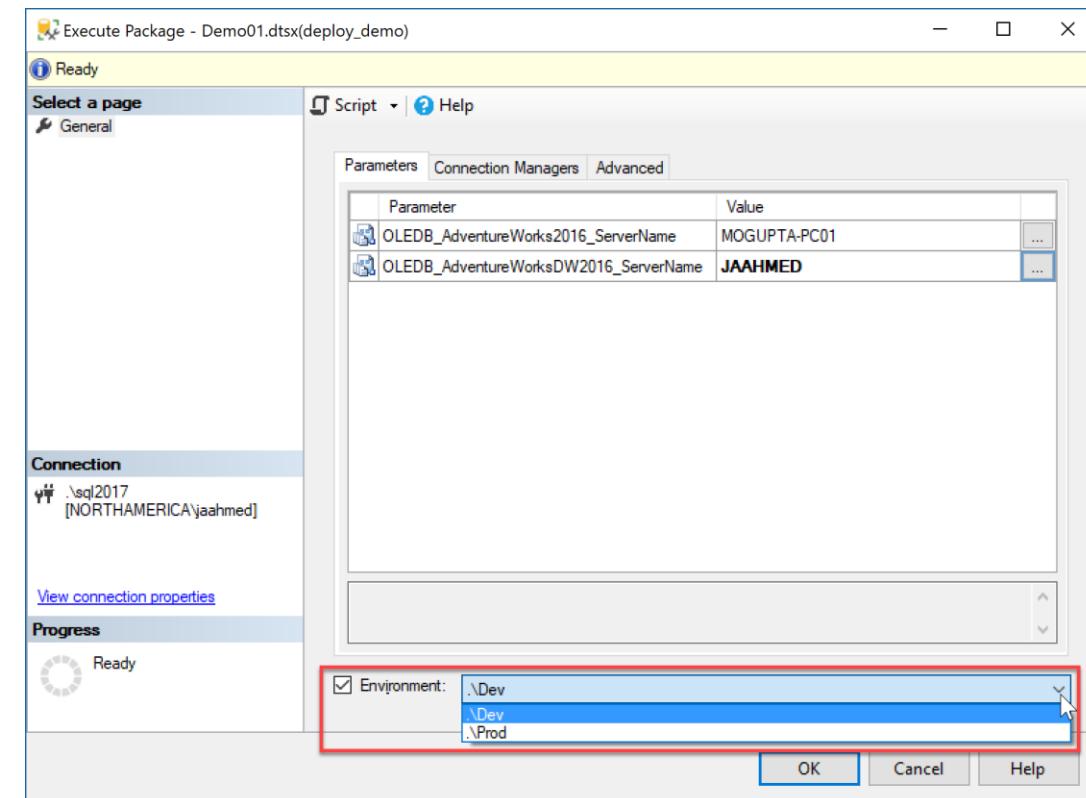
- Allows you to save a set of preconfigured values as an environment e.g.: Dev, QA, Prod
- Allows you to run the same package against different environments without creating multiple copies of the same project targeted for different servers (dev, qa, prod)

Environment Setup

- Create Environment, e.g.: Dev, QA, Prod
- Create Variables within each Environment
 - Preferably same name as project parameter or package parameter names for All environments.
- Configure Projects to reference Environment
 - i.e., which allowable environments can be used to configure the project
- Configure Mapping of Environment Variable to Parameter Value
 - Right-click Project -> Configure -> Parameters -> Set Parameter value to Environment Variable
- Validate Environment settings;
 - Right-click Project -> Validate

Executing using Environment setting

- Create Environment, e.g.: Dev, QA, Prod
- Environment Variable values will override default values



Deployment & Scheduling SSIS Jobs

Lab 01



Module 8: Best Practices

Module Overview

In this module, we will review

- SSIS Project Development Standards
- Review SSIS Design Strategies
- Review Logging Options
- Consider SSIS Security Options
- Configuration and Deployment Review

SSIS Project Development

Naming Conventions

- Important part of design and development to make life easier.
- Adopt a naming convention for packages, tasks and components.
- Proper naming and structure will keep packages, components, raw files, etc consistent across environments.
- Your package names should be clear in what the package does, while helping you organize them correctly.
- Create a folder structure for deployment and consistency
- Use environment in Projects in SSIS CatalogDB.

SSIS Package Development

Source Control

- Use the Source Control functionality in SSDT
 - Most source control system clients integrate into Visual Studio IDE
- It's critical to use source control software which provides standard version control functionality, and can scale to handle thousands of developers.
- If you use the Microsoft product stack for source control you can use one of the following:
 - Microsoft Visual SourceSafe
 - Microsoft Team Foundation Server

SSIS Package Development

Source Control

- Team Foundation Server and most source control applications have the following features:
 - Complete version control feature set.
 - Check-ins on a one change at a time basis.
 - Powerful branching and merging.
 - Shelving.
 - Check-in policies
- Team Foundation Server has the ability to structure an entire "data warehouse solution" as a TFS Project and to match the structure in Source Control
- Merging .dtsx files is not possible with TFS

SSIS Package Development

SSIS Connectivity

- A key areas of development in SQL Server Integration Services is improved options for connectivity.
- Integration Services provides:
 - A wide range of data source connectors out of the box
 - Many add-on connectors are available from Microsoft and from third-party vendors.
- The new connectivity options have also contributed to improving performance
- With many connectivity options available it can be difficult to know where to start.

SSIS Package Development

SSIS Connectivity

- Built-in connectors available with SSIS

OLE DB - Component Object Model (COM) interfaces that facilitate access to a variety of data stores

ADO.NET - ADO.NET source and destination options that are available in the data flow.

FLATFILE, MULTIFLATFILE, and FILE - Enables a package to access data in flat files

FTP and HTTP - Access FTP servers to transfer files and use HTTP connection manager to access a web server

MSMQ - Message Queue task uses the MSMQ connection manager

MSOLAP100 - OLE DB provider for SQL Server Analysis Services 10.0

SMSERVER - SMO connection managers used to perform transfer tasks

WMI - Windows Management Instrumentation (WMI) connection manager

XML - XML task and the XML source in the data flow.

SSIS Package Development

SSIS Connectivity

- Many more add-ons connectors can be install.
- These connectors are provided by Microsoft and others by third parties.
- Two main reasons why vendors create add-on connectors:
 - To facilitate access to a data source that is not supported by any of the built-in connectors
 - To provide an improvement in performance over existing connectors
- SSIS supports an ever-expanding range of connectivity options for a wide variety of data sources.

SSIS Package Development

SSIS Connectivity

- Add-ons Connectors

Microsoft Connector for Oracle by Attunity

Microsoft Connector for Teradata by Attunity

Microsoft Connector 1.0 for SAP BI

Adapters for SAP R/3 and Siebel eBusiness Applications (using adapters available in the BizTalk Adapter Pack 1.0)

Microsoft OLE DB Provider for DB2

- New connectors that have increased the emphasis on performance and the addition of ADO.NET data sources and destinations contribute to making SSIS a robust choice for ETL operations.

SSIS Package Development

Modularity

- Process Modularity
 - Break complex ETL into logically distinct packages (vs. monolithic design)
 - Improves development & debug experience
- Package Modularity
 - Separate sub-processes within package into separate Containers
 - More elegant, easier to develop
 - Simple to disable whole Containers when debugging
- Component Modularity
 - Use Script Task/Transform for one-off problems
 - Build custom components for maximum re-use

SSIS Design Review

Applications – Comparing Alternatives

- Key is identifying which application provides the right amount of functionality with the least amount of overhead.
- Example : BULK INSERT, BCP utility, SSIS which one should you use? TEST
- Compare the major functionality components of each application to see how they fit with your primary requirements.
- For a straight forward load with no data transformation requirements consider BCP or BULK INSERT.
- If you have multiple transformation requirements, or workflow management to control, SSIS is the way to go.

SSIS Design Review

Data Volume

- How much data must be processed?
- Reduce where possible
- Eliminate processing unwanted columns
- Conditional split for filtering rows
- Do not parse or convert columns unnecessarily
- In a fixed-width format you can combine adjacent unneeded columns into one
- Leave unneeded columns as strings

SSIS Design Review

Data Volume

- Use appropriate data types
- An integer in the range 1-999 takes 2 bytes as an integer, 3 bytes as a string, but 4 bytes as a real
- Suggest Types in the flat file connection manager UI
- Use parallelism
- If loading multiple files, can they be loaded in parallel?

SSIS Design Review

Data Profiling Task

- The Data Profiling task computes various profiles that help you become familiar with a data source and identify problems in the data that have to be fixed.
- You can use the Data Profiling task inside an Integration Services package to profile data that is stored in SQL Server and to identify potential problems with data quality.
- Creates a profile of SQL tables for exploring or maintaining data quality.
 - Run as a task in SSIS
 - Produces XML file output
 - Use the Data Profile Viewer to browse the xml output

SSIS Design Review

Data Profiling Task

- The Data Profiler Viewer lets you analyze a set of columns / tables looking for
 - Candidate keys
 - Column length distribution
 - Null Ratio
 - Pattern detection
 - Value distributions and stats
 - Functional dependencies
 - Value inclusion
- You can set up multiple profiles to be calculated then parse the data and analyze against a configured profile.

SSIS Design Review

T-SQL Merge Statement

- The SQL Merge statement allows you to merge the data in a result set with a table, based on a join condition that you specify.
- Why would using the MERGE statement be better than using the lookup pattern?
 - The data flow is simpler, With no branches in execution.
 - The data flow only has to write to a single table, instead of two tables.

SSIS Design Review

Using SQL CDC functionality with SSIS

- The Change Data Capture functionality can be taken advantage of in SSIS to do incremental loads.
- CDC captures changes to SQL Server tables (inserts, updates, deletes), and makes them available in an "easily-consumed, relational format".
- SSIS package can easily harvest the change data in the SQL Server databases to perform efficient incremental loads to a data warehouse.
- Change data capture is available only in SQL Server Enterprise, Developer, and Evaluation editions.

SSIS Design Review

Platform Considerations

- 32bit, x64 or I64 servers?
- Memory limits for 32bit Servers.
 - 2GB by default for a process on 32 bit servers
 - With the /3GB switch in boot.ini a process can access up to 3GB of memory
- Multiple SSIS packages run as separate process. They will each have its own virtual address space.
- 64bit Server Memory - 16TB-128TB for native process
- 4GB for 32bit process running on 64bit under WOW64

SSIS Design Review

Platform Considerations

- Running SQL and SSIS on the same server or running on separate server?
- SSIS and SQL are independent – on the same server they will compete for memory, CPU, and all resources.
- Best practice running SSIS on a separate server.
- Few specific scenarios when SSIS and SQL on the same server can have its advantages.
 - Consider a large load with limited transformations.
- Optimize and stabilize the basics. Disk, Memory, Windows, Network

SSIS Design Review

Database Considerations

- Database recovery model – options
- Simple, Bulk-logged, and Full.
- Pick the suitable logging for your business needs.
Simple and bulk-logged are desirable because it does less logging.
- Indexes are very important part of any design.
- Example: Taking advantage of existing indexes for search and sort operations can reduce much overhead.
- SQL 2005 onwards gives us the option to disable indexes during large load operations.

SSIS Design Review

Database Considerations

- Disabling indexes – Depends largely on the time it takes to rebuild the index afterwards.
- Filegroups - Consider placing large tables in its own file groups.
- Data Files – For busy databases consider multiple files for large tables.
- Partitioned tables can give a real boost to typical data warehouse loads if its done properly.

SSIS Design Review

Use Database Snapshots with SSIS

- A database snapshot is a read-only, static view of a database (the source database).
- Since each database snapshot is transactionally consistent with the source database as of the moment of the snapshot's creation. It can be used as the source system database to extract data.
- When data warehouse tables continue to change during an extract, there is a chance to extract out-of-sync data
- Snapshots can be used with SSIS to extract a consist view of the data from a given point in time.

SSIS Design Review

Use Database Snapshots with SSIS

1. Use a SQL Task and creating a snapshot of the source system database
 2. A Execute Script Task can switch the database ConnectionString from the original database to the newly created snapshot database.
 3. Use a sequence container and do all the data flow tasks.
 4. Second Script Task switches the ConnectionString back to the original database
 5. Final Execute SQL Task drops the database
-
- Database snapshots can be used with SSIS to create a resourceful combo for extracting consistent data.

SSIS Design Review

Or use SQL Server AlawaysOn Secondary

1. Configure AlwaysOn AG for Read-able secondaries.
2. Redirect all your read traffic to secondary to minimize impact to production servers.

SSIS Logging Options

- SSIS has event driven logging.
- SSIS enables logging to text files (most common), Windows event logs, profiler trace, and SQL Server.
- Log providers implement logging for packages, containers, and tasks
- Logs are critical for troubleshooting, auditing, and analysis.

SSIS Logging Options

SSIS Native Logging

- The Text File log provider, which writes log entries to ASCII text files in a comma-separated value (CSV) format (log).
- The SQL Server Profiler log provider, which writes traces that you can view using SQL Server Profiler (.trc).
- The Windows Event log provider, which writes entries to the Application log in the Windows Event log on the local computer.
- The XML File log provider, which writes log files to an XML file (.xml).

SSIS Logging Options

Extended Logging

- SSIS lets you create custom log formats.
- User has the ability to override the base log format with custom logging.
- SSIS also log to multiple formats or filter specific events to specific logs.

SSIS Logging Options

DTEXEC.EXE /VLOG option

- Add logging at the command line without editing package in design mode.
- To have Integration Services enable a log provider for text files and write log events to a specified text file, include a path and file name as the Filespec parameter.
 - >DTEXEC.EXE /VLOG [Filespec]
 - >DTEXEC.EXE /VLOG c:\MyPackage\Log\mylog.txt
- Beware that it collects all diagnostic events. Could slow down the package

SSIS Logging Options

Beyond logging – Using improved memory dumps

- Use enhanced memory dump capability for debugging crashes and hung packages.
- When a dump is created (the directory is configurable, but defaults to %ProgramFiles%\Microsoft SQL Server\140\Shared\ErrorDumps).
- We also generate a text file which contains useful debugging information that SSIS developers and support can make use of.
- These dumps will be created automatically during a crash, but can also be triggered by the user.

SSIS Logging Options

Beyond logging – Using improved memory dumps

- Dumping with DTExec
 - Two dumping options have been added to **DTExec - /Dump**, which takes a semi-colon separated list of error codes (HResults) to dump on, and **/DumpOnError**, which will trigger a memory dump anytime an error is encountered.
- Dumping with DTUtil
 - The **/Dump** option added to **DTUtil** allows you to create a memory dump for a running package without disrupting its execution (at least not for very long). This is very useful if you suspect your package has hung.
- When you specify a debug dump option, Integration Services creates the following debug dump files:
 - A .mdmp debug dump file. This is a binary file.
 - The .tmp debug dump file. This is a text formatted file

SSIS Security

- Security is comprised of several layers to support both SQL and File System Based scenarios.
 - Packages can be Encrypted
 - Packages can be Digitally Signed
 - Packages can be stored in SQL DB and protected with SQL Roles
- SSIS packages can be signed with a certificate and the package can be configured to require the runtime to check the signature before loading the package
- SSIS includes the three fixed database-level roles db_ssisadmin, db_ssisltduser, and db_ssisisoperator for controlling access to packages.

Deployment Review

- Use Package Environments
 - Build it in from the start will make things easier later on
 - Simplify deployment Dev -> QA -> Production
- Use Package Logging
 - If you have very complex package, remove detailed logging before deploying to production (Use minimal logging).
 - Logging does have overhead.
- Build in Security from the start
 - Credentials and other sensitive info
 - Package & Process IP
 - Configurations & Parameters

Development and Deployment Tips

- Use source code control system
- Undo is not as simple in a GUI environment!
- Comment your packages and scripts
- In 2 weeks even you may forget a subtlety of your design
someone else may have to maintain your code
- Use error-handling
- Use the correct precedence constraints on tasks
- Use the error outputs on transforms – store them in a
table for processing later, or use downstream if the error
can be handled in the package
- Use Try...Catch in your scripts

