

# ***SQL Server Integration Services (SSIS) Training Kit (PART 3)***

Lesson 10: Operating system-level  
tasks

## File System Task

- “The File System task performs operations on files and directories in the file system”
- For example, by using the File System task, a package can create, move, or delete directories and files. You can also use the File System task to set attributes on files and directories. For example, the File System task can make files hidden or read-only.
- All File System task operations use a source, which can be a file or a directory. The source can be specified by using a File connection manager that points to the directory or file or by providing the name of a variable that contains the source path.
- The operations that copy and move file and directories and rename files use a destination and a source. The destination is specified by using a File connection manager or a variable. File system task operations can be configured to permit overwriting of destination files and directories.



Copyright © Capgemini 2015. All Rights Reserved 2

## File System Task

- The File System task includes a predefined set of operations. The following table describes these operations.

Operation	Description
Copy directory	Copies a folder from one location to another.
Copy file	Copies a file from one location to another.
Create directory	Creates a folder in a specified location.
Delete directory	Deletes a folder in a specified location.
Delete directory content	Deletes all files and folders in a folder.
Delete file	Deletes a file in a specified location.
Move directory	Moves a folder from one location to another.
Move file	Moves a file from one location to another.
Rename file	Renames a file in a specified location.
Set attributes	Sets attributes on files and folders. Attributes include Archive, Hidden, Normal, Read only, and System. Normal is the lack of attributes, and it cannot be combined with other attributes. All other attributes can be used in combination.



## File System Task

### ▪ Is Destination Path Variable

- Indicate whether the destination path is stored in a variable. This property has the options listed in the following table.
  - True - The destination path is stored in a variable.
  - False - The destination path is specified in a File connection manager.

### ▪ Overwrite Destination

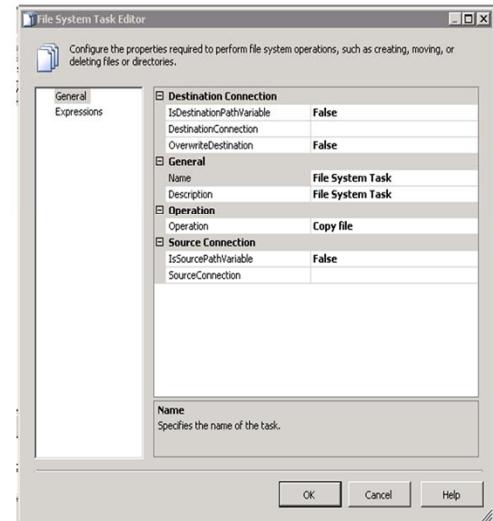
- Specify whether the operation can overwrite files in the destination directory.

### ▪ Name

- Provide a unique name for the File System task. This name is used as the label in the task icon. Task names must be unique within a package.

### ▪ Description

- Type a description of the File System task.



Copyright © Capgemini 2015. All Rights Reserved 4

## File System Task

### ■ Operation

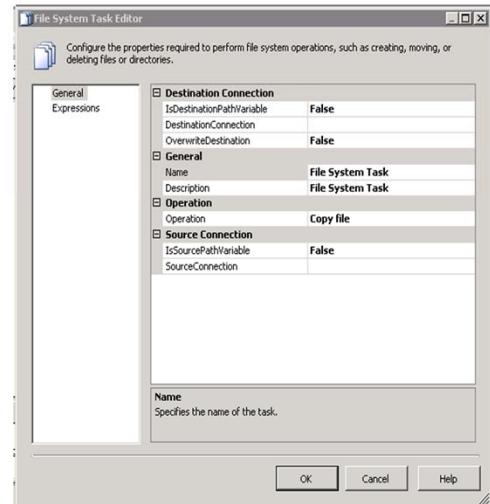
- Select the file-system operation to perform. This property has the options listed in the following table.

### ■ Is Source Path Variable

- Indicate whether the destination path is stored in a variable. This property has the options listed in the following.
  - True - The destination path is stored in a variable.
  - False - The destination path is specified in a File connection manager.

### ■ Description

- Type a description of the File System task.



## FTP Task

- The FTP task downloads and uploads data files and manages directories on servers.
- FTP task can be for the following purposes:

Creates a folder on the FTP server.

Sends a file from the local computer to the FTP server.

Saves a file from the FTP server to the local computer.

Deletes a folder on the local computer.

Deletes a folder on the FTP server.

Deletes a file on the local computer.

Deletes a file on the FTP server.



Copyright © Capgemini 2015. All Rights Reserved 6

The FTP Task works very similarly to any FTP utility. It allows us to send and receive files from an FTP location along with other FTP commands.

## Send Mail Task

- The Send Mail Task sends email via Simple Mail Transfer Protocol (SMTP).
- The Send Mail Task is most commonly used as a notification tool.
- This allows to receive information about the package that can be passed into the mail task through variables.
- The Send Mail Task can be placed at the end of a Control Flow to send email on the successful
- The Send Mail Task can also be used to send files, because it has the ability to send attachments.



## Send Mail Task

- Can configure the Send Mail task in the following ways:

Provide the message text for the e-mail message.

Provide a subject line for the e-mail message.

Set the priority level of the message.

Specify the recipients on the To, Cc, and Bcc lines.



Copyright © Capgemini 2015. All Rights Reserved 8

## SQL Server Tasks

Lesson 11:

## Bulk Insert Task

- The Bulk Insert task provides an efficient way to copy large amounts of data into a SQL Server table or view.
- It can transfer data only from a text file into a SQL Server table or view.
- The destination must be a table or view in a SQL Server database.
- If the text file is located on the same computer as the SQL Server database into which data is inserted, the copy operation occurs at an even faster rate because the data is not moved over the network.



Copyright © Capgemini 2015. All Rights Reserved 10

The Bulk Insert task provides an efficient way to copy large amounts of data into a SQL Server table or view. For example, suppose your company stores its million-row product list on a mainframe system, but the company's e-commerce system uses SQL Server to populate Web pages. You must update the SQL Server product table nightly with the master product list from the mainframe. To update the table, you save the product list in a tab-delimited format and use the Bulk Insert task to copy the data directly into the SQL Server table.

The Bulk Insert task can transfer data only from a text file into a SQL Server table or view. To use the Bulk Insert task to transfer data from other database management systems (DBMSs), you must export the data from the source to a text file and then import the data from the text file into a SQL Server table or view.

## Execute SQL Task

The Execute SQL task runs SQL statements or stored procedures from a package. The task can contain either a single SQL statement or multiple SQL statements that run sequentially.

- Execute SQL task can be used for the following purposes:
- Truncate a table or view in preparation for inserting data.
- Create, alter, and drop database objects such as tables and views.
- Re-create fact and dimension tables before loading them.
- Run stored procedures.
- Save the row set returned from a query into a variable.



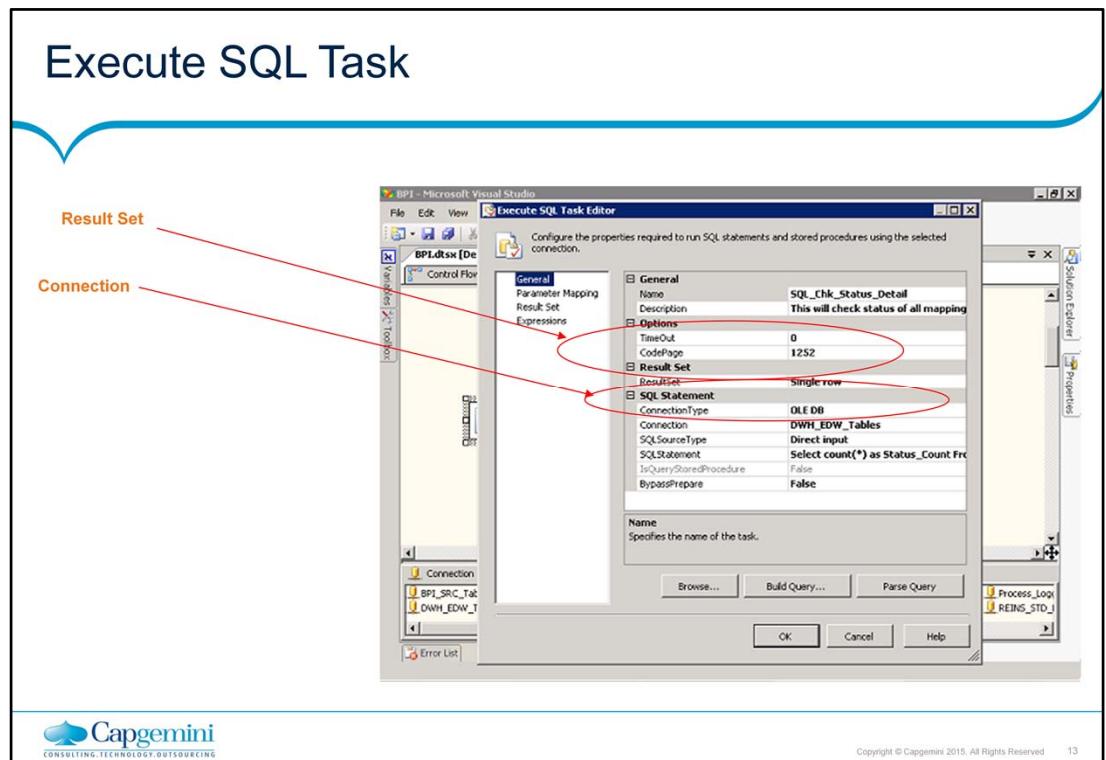
Copyright © Capgemini 2015. All Rights Reserved 11

## Execute SQL Task

- Specify the type of connection manager to use to connect to a database.
- Specify the type of result set that the SQL statement returns.
- Specify a time-out for the SQL statements.
- Specify the source of the SQL statement.
- Indicate whether the task skips the prepare phase for the SQL statement. If using the ADO connection type, indicate whether the SQL statement is a stored procedure. For other connection types this property is read-only and its value is always false.



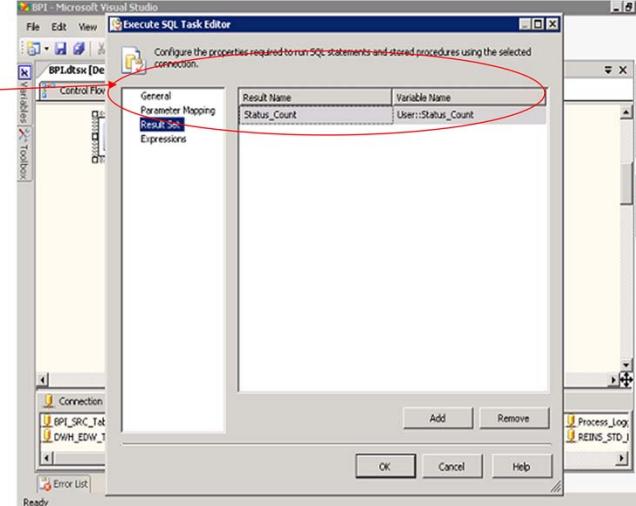
Copyright © Capgemini 2015. All Rights Reserved 12



## Execute SQL Task

Defining Result Set

- Specify Result Name
- Assign a variable



Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 14

## Transfer Database Task

- The Transfer Database task transfers a SQL Server database between two instances of SQL Server.
- This task supports two modes of database transfer - namely DatabaseOnline and DatabaseOffline.
- The Transfer Database task can also be configured to permit overwriting a destination database that has the same name, replacing the destination database.



Copyright © Capgemini 2015. All Rights Reserved 15

The Transfer Database task transfers a SQL Server database between two instances of SQL Server. In contrast to the other tasks that only transfer SQL Server objects by copying them, the Transfer Database task can either copy or move a database. The task can copy a database between instances of SQL Server 2000, instances of SQL Server, or one of each. This task can also be used to copy a database within the same server. The database can be transferred by using online or offline mode. When you use online mode, the database remains attached and it is transferred by using SQL Management Object (SMO) to copy the database objects. When you use offline mode, the database is detached, the database files are copied or moved, and the database is attached at the destination after the transfer finishes successfully. If the database is copied, it is automatically reattached at the source if the copy is successful. In offline mode, the database is copied more quickly, but the database is unavailable to users during the transfer.

Offline mode requires that you specify the network file shares on the source and destination servers that contain the database files. If the folder is shared and can be accessed by the user, you can reference the network share using the syntax `\computername\Program Files\myfolder\`. Otherwise, you must use the syntax `\computername\c$\Program Files\myfolder\`. To use the latter syntax, the user must have write access to the source and destination network shares.

## Transfer Error Messages Task

- The Transfer Error Messages task transfers one or more SQL Server user-defined error messages between instances of SQL Server.
- User-defined messages are messages with an identifier that is equal to or greater than 50000
- The Transfer Error Messages task can be configured to transfer all error messages, or only the specified error messages
- An error message is defined as a duplicate error message if the identifier and the language are the same.
- The Transfer Error Messages task can be configured to handle existing error messages in the following ways:

Overwrite existing jobs.

Fail the task when duplicate jobs exist.

Skip duplicate jobs.



Copyright © Capgemini 2015. All Rights Reserved 16

The Transfer Error Messages task transfers one or more SQL Server user-defined error messages between instances of SQL Server. User-defined messages are messages with an identifier that is equal to or greater than 50000. Messages with an identifier less than 50000 are system error messages, and cannot be transferred by using the Transfer Error Messages task.

The Transfer Error Messages task can be configured to transfer all error messages, or only the specified error messages. User-defined error messages may be available in a number of different languages and the task can be configured to transfer only messages in selected languages.

The **sysmessages** table in the **master** database contains all the error messages—both system and user-defined—that SQL Server uses.

The user-defined messages to be transferred may already exist on the destination. An error message is defined as a duplicate error message if the identifier and the language are the same. The Transfer Error Messages task can be configured to handle existing error messages in the following ways:

Overwrite existing error messages.

Fail the task when duplicate messages exist.

Skip duplicate error messages.

## Transfer Jobs Task

- The Transfer Jobs task transfers one or more SQL Server Agent jobs between instances of SQL Server.
- The Transfer Jobs task can be configured to transfer all jobs, or only specified jobs. You can also indicate whether the transferred jobs are enabled at the destination.
- The jobs to be transferred may already exist on the destination. The Transfer Jobs task can be configured to handle existing jobs in the following ways:

Overwrite existing jobs.

Fail the task when duplicate jobs exist.

Skip duplicate jobs.



Copyright © Capgemini 2015. All Rights Reserved 17

We can use SQL Server Integration Services to transfer the jobs from SQL 2005 to another SQL 2005 or SQL 2008.

## Transfer Login Task

- The Transfer Logins task transfers one or more logins between instances of SQL Server.
- The Transfer Logins task supports a source and destination that is SQL Server 2000 or SQL Server..
- The Transfer Logins task can be configured to transfer all logins, only specified logins, or all logins that have access to specified databases only.
- The logins to be transferred may already exist on the destination. The Transfer Logins task can be configured to handle existing logins in the following ways:

Overwrite existing jobs.



Fail the task when duplicate jobs exist.



Skip duplicate jobs.



Copyright © Capgemini 2015. All Rights Reserved 18

We can use SQL Server Integration Services to **Transfer Login** from SQL 2005 to another SQL 2005 or SQL 2008 using **Transfer Login Task**.

The Transfer Logins task supports a source and destination that is SQL Server 2000 or SQL Server. There are no restrictions on which of the two versions to use as a source or destination.

## Transfer Master Stored Procedures Task & Transfer SQL Server Objects Task

### ▪ Transfer Master Stored Procedures Task

- The Transfer Master Stored Procedures task transfers one or more user-defined stored procedures between master databases on instances
- The Transfer Master Stored Procedures task can be configured to transfer all stored procedures or only specified stored procedures. This task does not copy system stored procedures.

### ▪ Transfer SQL Server Objects Task

- The Transfer SQL Server Objects task transfers one or more types of objects in a SQL Server database between instances of SQL Server.



Copyright © Capgemini 2015. All Rights Reserved 19

The Transfer Master Stored Procedures task transfers one or more user-defined stored procedures between **master** databases on instances of SQL Server. To transfer a stored procedure from the **master** database, the owner of the procedure must be **dbo**.

The Transfer Master Stored Procedures task can be configured to transfer all stored procedures or only specified stored procedures. This task does not copy system stored procedures.

As other transfer task in **Transfer Master Stored Procedures Task**

The master stored procedures to be transferred may already exist on the destination. The Transfer Master Stored Procedures task can be configured to handle existing stored procedures in the following ways:

Overwrite existing stored procedures.

Fail the task when duplicate stored procedures exist.

Skip duplicate stored procedures.

**Transfer SQL Server Objects Task**

The Transfer SQL Server Objects task transfers one or more types of objects in a SQL Server database between instances of SQL Server. For example, the task can copy tables and stored procedures. Depending on the version of SQL Server that is used as a source, different types of objects are available to copy. For example, only a SQL Server database includes schemas and user-defined aggregates.

Like other Transfer task **Transfer SQL Server Objects Task**

can be configured to transfer all objects, all objects of a type, or only specified objects of a type. For example, you can choose to copy only selected tables in the **AdventureWorks** database.

And if the Transfer SQL Server Objects task transfers tables, you can specify the types of table-related objects to copy with the tables. For example, you can specify that primary keys are copied with tables.

List of objects those can be transfer are:

Tables

Views

Stored Procedures

User-Defined Functions

Defaults

User-Defined Data Types

Partition Functions

Partition Schemes

Schemas

Assemblies

User-Defined Aggregates

User-Defined Types

XML Schema Collection

## Scripting Tasks

Lesson 12:

## Scripting Tasks

- Script Task
- The Script task provides code to perform functions that are not available in the built-in tasks and transformations that SQL Server Integration Services provides.
- This task can accomplish anything that can be done with any .NET programming. The Script task can also combine functions in one script instead of using multiple tasks and transformations.



Copyright © Capgemini 2015. All Rights Reserved 21

When we create a new SQL Server Integration Services (SSIS) package we find ourselves wanting some functionality that the built-in tasks cannot accomplish. This situation is where the Script Task comes into play.

## Data Profiling Tasks

Lesson 13:

## Data Profiling Task

- “The Data Profiling Task The task lets you analyze data in a SQL Server database and, from the results of that analysis, generate XML reports that can be saved to a file or an SSIS variable ”
- By configuring one or more of the task's profile types, you can generate a report that provides details such as a column's minimum and maximum values, or the number and percentage of null values.
- By accessing the results generated by the Data Profiling task, you can design a workflow that automatically determines the appropriate actions to take based on the validity of the source data.

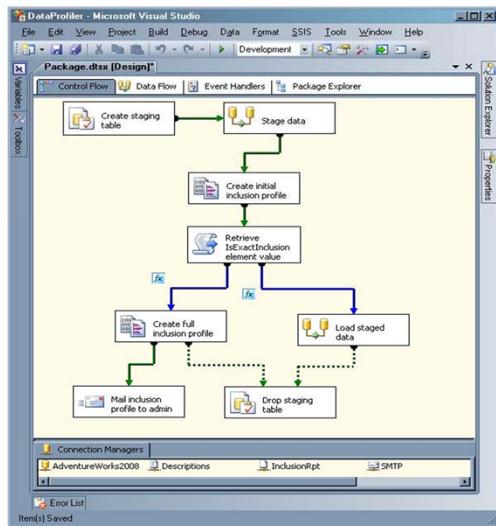


Copyright © Capgemini 2015. All Rights Reserved 23

## Data Profiling Task

### Example

- For our example data profiling task, let's suppose we're extracting product descriptions (such as colors, sizes, and other details) from a text file and want to ensure that the product IDs in that file match the product IDs in the target database before loading the description data. We can stage the data in a SQL Server table and then use the Data Profiling task to generate a Value Inclusion report, which will tell us whether the source IDs are valid. After we determine the data's validity, we can then take the appropriate actions within our SSIS package. Figure shows what such a solution might look like, as it appears in the Control Flow tab of SSIS Designer (in Business Intelligence Development Studio).



## Data Profiling Task

### Solution

- In this solution, we first stage the data because the Data Profiling task requires that the data be in a SQL Server database in order to perform the analysis. Next, we generate the Value Inclusion report (saved to a variable), and then use a Script task to extract the analysis results from that variable. If one or more IDs are invalid, we generate the report as an XML document and email it to an administrator. If all the IDs are valid, we load the data into the target database



Copyright © Capgemini 2015. All Rights Reserved 25

## Data Profiling Task

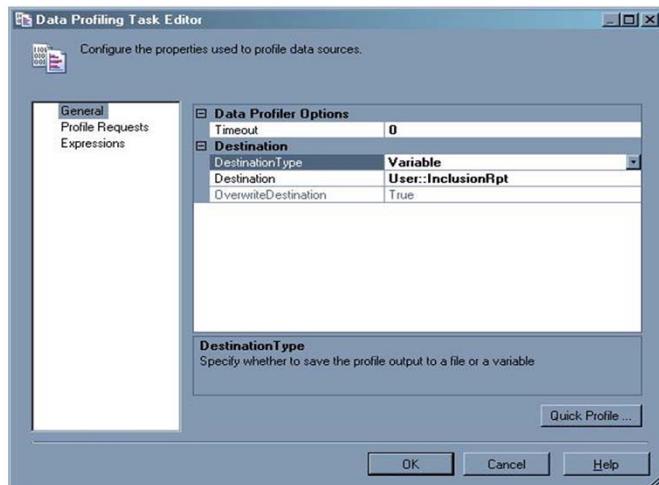
### ■ Generating the Inclusion Profile

- The next step is to add the Data Profiling task and connect the precedence constraint from the Data flow task to the Data Profiling task. When you set up a Data Profiling task, you can configure it to save data to an XML file or to an SSIS string variable. In this case, I want to save the report information to a variable so I can later use that variable in the Script task.
- Ultimately, the package will require two string variables—one to hold the report, and a second one to hold the analysis results retrieved through the Script task (this process is explained later in the article). So before configuring the Data Profiling task, we need to create these two string variables, both of which will be configured at the package scope:
  - 1.InclusionRpt, to hold the Value Inclusion report
  - 2.Results, to hold the analysis results retrieved through the Script task.
- When configuring the Data Profiling task, select Variable as the destination type and then specify the InclusionRpt variable as the destination.



Copyright © Capgemini 2015. All Rights Reserved 26

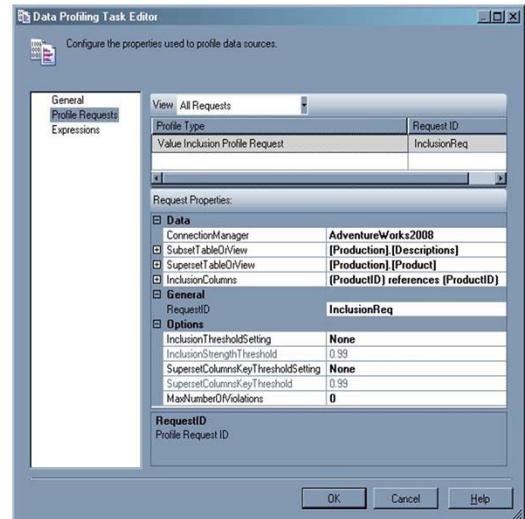
## Data Profiling Task



Creating the InclusionRpt Variable

## Data Profiling Task

- After specifying our variable, we can now start configuring the Value Inclusion profile.
- Click to the Profile Requests page, and select Value Inclusion Profile Request in the Profile Type drop-down list (in the first cell of the grid). You can then configure the data profile in the bottom section of the Profile Requests page, as shown in the Figure :



## Workflow Tasks

Lesson 14:

## Execute Package Task

- Execute Package Task can run other packages as part of a workflow.
- A package that runs other packages is generally referred to as the parent package, and the packages that a parent workflow runs are called child packages.
- The child package can be run in the process of the parent package, or it can be run in its own process.
- The Execute Package task can run packages stored in the SQL Server msdb database and packages stored in the file system.
- Can use the Execute Package task for the following purposes:



Copyright © Capgemini 2015. All Rights Reserved 30

The Execute Package task extends the enterprise capabilities of Integration Services by letting packages run other packages as part of a workflow.

You can use the Execute Package task for the following purposes:

Breaking down complex package workflow. This task lets you break down workflow into multiple packages, which are easier to read, test, and maintain. For example, if you are loading data into a star schema, you can build a separate package to populate each dimension and the fact table.

Reusing parts of packages. Other packages can reuse parts of a package workflow. For example, you can build a data extraction module that can be called from different packages. Each package that calls the extraction module can perform different data scrubbing, filtering, or aggregation operations.

Grouping work units. Units of work can be encapsulated into separate packages and joined as transactional components to the workflow of a parent package. For example, the parent package runs the accessory packages, and based on the success or failure of the accessory packages, the parent package either commits or rolls back the transaction.

Controlling package security. Package authors require access to only a part of a multipackage solution. By separating a package into multiple packages, you can provide a greater level of security, because you can grant an author access to only the relevant packages.

## Execute Package Task

Execute Package Task uses

Breaking down  
complex  
package  
workflow.

Reusing parts of  
packages.

Grouping work  
units

Controlling  
package  
security



Copyright © Capgemini 2015. All Rights Reserved 31

## Workflow Tasks.....

- Execute Process Task
- The Execute Process task runs an application or batch file as part of a SQL Server Integration Services package workflow.
- Execute Process task can open any standard application.
- When the Execute Process task runs a command-line application, it provides input to the application through a variable
- Message Queue Task
- The Message Queue task allows you to use Message Queuing (also known as MSMQ) to send and receive messages between SQL Server Integration Services packages, or to send messages to an application queue that is processed by a custom application.



Copyright © Capgemini 2015. All Rights Reserved 32

The Execute Process task runs an application or batch file as part of a SQL Server Integration Services package workflow. Although we can use the Execute Process task to open any standard application, such as Microsoft Excel or Microsoft Word, you typically use it to run business applications or batch files that work against a data source. For example, you can use the Execute Process task to expand a compressed text file. Then the package can use the text file as a data source for the data flow in the package. As another example, you can use the Execute Process task to run a custom Visual Basic application that generates a daily sales report.

### Message Queue Task

You can use the Message Queue task for the following purposes:  
Delaying task execution until other packages check in. For example, after nightly maintenance at each of your retail sites, a Message Queue task sends a message to your corporate computer. A package running on the corporate computer contains Message Queue tasks, each waiting for a message from a particular retail site. When a message from a site arrives, a task uploads data from that site. After all the sites have checked in, the package computes summary totals.

Sending data files to the computer that processes them. For example, the output from a restaurant cash register can be sent in a data file message to the corporate payroll system, where data about each waiter's tips is extracted.

Distributing files throughout your enterprise. For example, a package can use a Message Queue task to send a package file to another computer. A package running on the destination computer then uses a Message Queue task to retrieve and save the package locally.

## Workflow Tasks.....

- 
- Message Queue Uses
  - Delaying task execution until other packages check in
  - Sending data files to the computer that processes them
  - Distributing files throughout your enterprise

# Processing XML

Lesson 15:

## XML Task

- The XML task can retrieve XML documents, apply operations to the documents, merge multiple documents, or validate, compare, and save the updated documents to files and variables.



Copyright © Capgemini 2015. All Rights Reserved 35

The Web Service task executes a Web service method.

Web Service task can be used for the following purposes:

Writing to a variable the values that a Web service method returns. For example, you could obtain the highest temperature of the day from a Web service method, and then use that value to update a variable that is used in an expression that sets a column value.

Writing to a file the values that a Web service method returns. For example, a list of potential customers can be written to a file and the file then used as a data source in a package that cleans the data before it is written to a database.

## **Analysis Services Tasks**

Lesson 16:

## Analysis Services Tasks

- Analysis Services Tasks within SSIS can be used to create, modify, delete and process Analysis Services objects.
- The different types of Analysis Services tasks are as follows:
- Analysis Services Execute DDL Task : It is similar to the Execute SQL Task, however using the Analysis Services Execute DDL Task you can issue Data Definition Language statements against an Analysis Services system. The DDL statements can be used to create cubes, dimensions, KPI's or any other analytical processing of OLAP objects.
- The Analysis Services Processing Task : It can be used to process analysis services objects such as cubes, dimensions and mining models.
- The Data Mining Query Task : It can be used to run prediction queries based on data mining models built into analysis services. A prediction query creates a prediction for new data by using the mining model chosen.



Copyright © Capgemini 2015. All Rights Reserved 37

## Maintenance Tasks

Lesson 17 :

## Maintenance Tasks

- SQL Server Integration Services includes a set of tasks that perform database maintenance functions. These tasks are commonly used in database maintenance plans, but the tasks can also be included in SSIS packages.



## Execute T-SQL Statement Task and Update Statistics Task

- Execute T-SQL statement task
- This task is similar to the Execute SQL Task, however it only supports Transact SQL Statements. It should be used for SQL Server specific SQL statements.
- You cannot use this task to run statements on servers that use other dialects of the SQL language. If you need to run parameterized queries, save the query results to variables, or use property expressions, you should use the Execute SQL task instead of the Execute T-SQL Statement task.
- Update Statistics Task
- This task will update the statistics for one of many tables in one or many databases.



Copyright © Capgemini 2015. All Rights Reserved 40

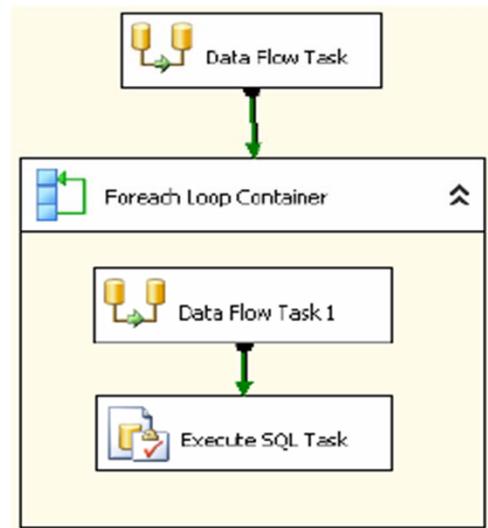
## Container

Lesson 18:

## ForEach Loop Container

INDEX

- SQL Server 2008 Integration Services (SSIS) includes the For each Loop container, a control flow element that makes it simple to include a looping construct that enumerates files and objects in the control flow of a package.
- The For each Loop container provides no functionality; it provides only the structure in which you build the repeatable control flow, specify an enumerator type, and configure the enumerator. To provide container functionality, you must include at least one task in the For each Loop container.
- The Foreach Loop container can include a control flow with multiple tasks and other containers. Adding tasks and containers to a Foreach Loop container is similar to adding them to a package, except you drag the tasks and containers to the Foreach Loop container instead of to the package. If the Foreach Loop container includes more than one task or container, you can connect them using precedence constraints just as you do in a package.



## For Each Loop Container

SQL Server 2008 Integration Services (SSIS) provides the following enumerator types:

- For Each ADO enumerator to enumerate rows in tables. For example, you can get the rows in an ADO record set.
- For Each ADO.NET Schema Row set enumerator to enumerate the schema information about a data source. For example, you can enumerate and get a list of the tables in the Adventure Works SQL Server database.
- For Each Node list enumerator to enumerate the result set of an XML Path Language (XML Path) expression. For example, this expression enumerates and gets a list of all the authors in the classical period:  
`/authors/author[@period='classical']`.
- For Each SMO enumerator to enumerate SQL Server Management Objects (SMO) objects. For example, you can enumerate and get a list of the views in a SQL Server database.



## For Each Loop Container

- For each File enumerator to enumerate files in a folder. The enumerator can traverse subfolders. For example, you can read all the files that have the \*.log file name extension in the Windows folder and its subfolders.
- For Each From Variable enumerator to enumerate the enumerable object that a specified variable contains. The enumerable object can be an array, an ADO.NET Data Table, an Integration Services enumerator, and so on. For example, you can enumerate the values of an array that contains the name of servers.
- For Each Item enumerator to enumerate items that are collections. For example, you can enumerate the names of execu tables and working directories that an Execute Process task uses.

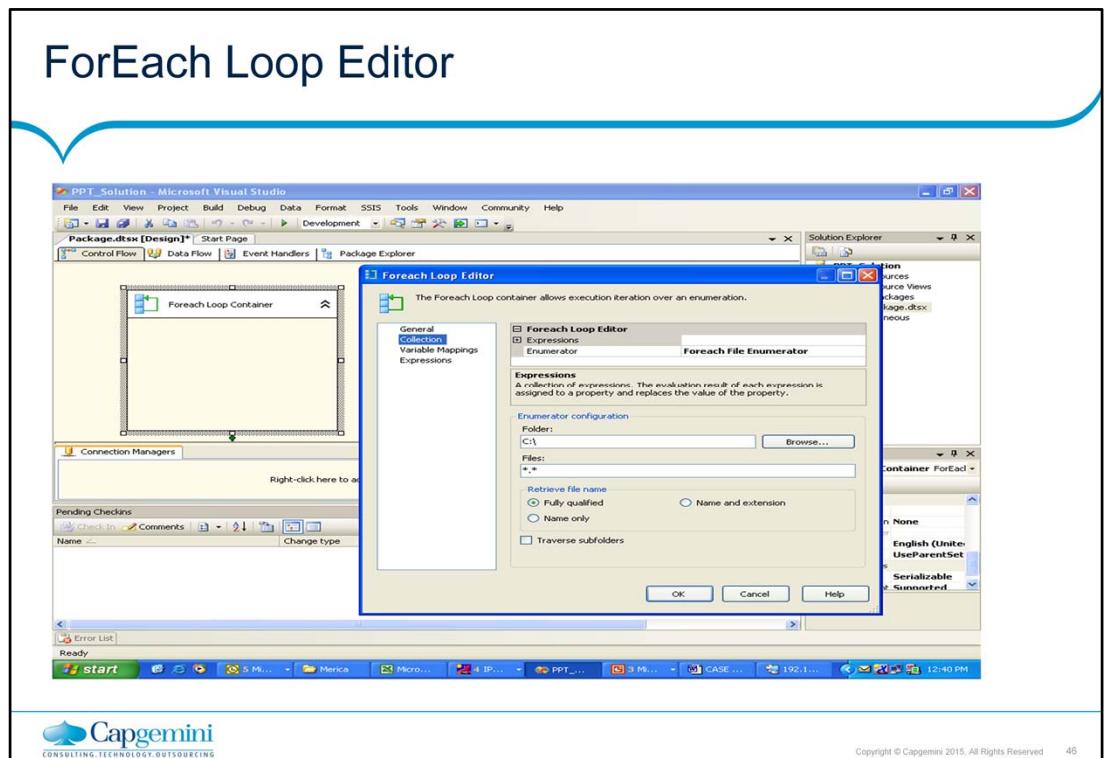


Copyright © Capgemini 2015. All Rights Reserved 44

## ForEach Loop Container

Enumerator	Configuration requirements
ForEach ADO	Specify the ADO object source variable and the enumerator Mode.
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 Node list	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 Loop Editor

- Use the General page of the For each Loop Editor dialog box to name and describe a For each Loop container that uses a specified enumerator to repeat a workflow for each member in a collection.
- Use the Collection page of the For each Loop Editor dialog box to specify the enumerator type and configure the enumerator.
- Use the Variables Mappings page of the For each Loop Editor dialog box to map variables to the collection value. The value of the variable is updated with collection values on each iteration of the loop.
- Lets look at the two most commonly used enumerator types:
- For Each File Enumerator
- For Each ADO Enumerator
- 



Copyright © Capgemini 2015. All Rights Reserved 47

## ForEach File Enumerator

- In case of 'File enumerator' – Specify the folder path, the for each loop will run for the number of files in this folder.
- If the 'files' option: \*.\* will execute the for each loop for all files.  
\*.txt will execute the for each loop for all text files.
- Fully qualified
  - Select to retrieve the fully qualified path of file names.
- Name only
  - Select to retrieve only the file names.
- Name and extension
  - Select to retrieve the file names and their file name extensions.
  - Traverse Subfolders
  - Select to include the subfolders in the enumeration.



Copyright © Capgemini 2015. All Rights Reserved 48

## ForEach ADO Enumerator

- ADO object source variable
  - Select a user-defined variable in the list, or click <New variable...> to create a new variable. The variable must have the Object data type, otherwise an error occurs.
  - Rows in first table
  - Select to enumerate only rows in the first table.
- Rows in all tables (ADO.NET dataset only)
  - Select to enumerate rows in all tables. This option is available only if the objects to enumerate are all members of the same ADO.NET dataset.
- All tables (ADO.NET dataset only)
  - Select to enumerate tables only.



Copyright © Capgemini 2015. All Rights Reserved 49

## For loop and sequence container

- For loop

- The For Loop container defines a repeating control flow in a package. In each repeat of the loop, the For Loop container evaluates an expression and repeats its workflow until the expression evaluates to False.
- The For Loop container uses the following elements to define the loop:
- An optional initialization expression that assigns values to the loop counters.
- An evaluation expression that contains the expression used to test whether the loop should stop or continue.
- An optional iteration expression that increments or decrements the loop counter.
- Sequence container
- Sequential Task groups the tasks into multiple control flows and executes the process sequentially. We can use this task widely based on our requirement like, disabling a sequence when it should not process, use it when managing multiple tasks at a same time in one location.



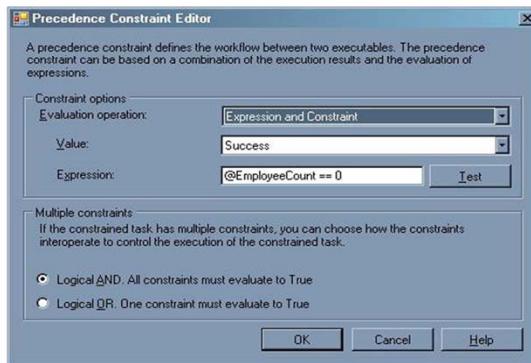
Copyright © Capgemini 2015. All Rights Reserved 50

## Precedence Constraint

Lesson 19:

## Configuring the Precedence Constraint

- Besides the regular Success and Failure constraints, can define precedence constraint Workflow by Expressions
- To add an expression, double-click the precedence constraint to open the Precedence Constraint Editor dialog box, as shown in below figure.



## Configuring the Precedence Constraint..

- To add an expression to a precedence constraint, the first step is to select Evaluation operation. And Evaluation operation has following options:

### Constraint:

- The precedence constraint is evaluated solely on the option selected in the Value property. The precedence constraint will evaluate to true only if the precedence executable runs successfully.

### Expression:

- The precedence constraint is evaluated based on the expression defined in the Expression text box. If the expression evaluates to true, the workflow continues and the constrained executable runs.

### Expression and Constraint:

- The precedence constraint is evaluated based on both the Value property and the expression. Both must evaluate to true for the constrained executable to run.

### Expression or Constraint:

- The precedence constraint is evaluated based on either the Value property or the expression. At least one of these properties must evaluate to true for the constrained executable to run.



Copyright © Capgemini 2015. All Rights Reserved 53

To add an expression to a precedence constraint, the first step is to select Evaluation operation. And Evaluation operation has following options:

**Constraint:** The precedence constraint is evaluated solely on the option selected in the Value property. For example, if Constraint is selected as the Evaluation operation option and Success is selected as the Value option (the default settings for both properties), the precedence constraint will evaluate to true only if the precedence executable runs successfully. When the precedence constraint evaluates to true, the workflow continues and the constrained executable runs. When the Constraint option is selected, the Expression property is greyed out.

**Expression:** The precedence constraint is evaluated based on the expression defined in the Expression text box. If the expression evaluates to true, the workflow continues and the constrained executable runs. If the expression evaluates to false, the constrained executable does not run. When the Expression option is selected, the Value property is greyed out.

**Expression and Constraint:** The precedence constraint is evaluated based on both the Value property and the expression. Both must evaluate to true for the constrained executable to run.

**Expression or Constraint:** The precedence constraint is evaluated based on either the Value property or the expression. At least one of these properties must evaluate to true for the constrained executable to run.

Define precedence constraint Workflow by Logical AND or Logical OR

**Precedence Constraint Editor** dialog box has following two options at the bottom :

**Logical AND:** All precedence constraints that point to the constrained executable must evaluate to true in order for that executable to run. This is the default option. If it is selected, the arrow is solid.

**Logical OR:** Only one precedence constraint that points to the constrained executable must evaluate to true in order for that executable to run. If this option is selected, the arrow is dotted.

## Configuring the Precedence Constraint..

- Define precedence constraint Workflow by Logical AND or Logical OR:
- Precedence Constraint Editor dialog box has following two options at the bottom:

**Logical AND**

- All precedence constraints that point to the constrained executable must evaluate to true in order for that executable to run. This is the default option. If it is selected, the arrow is solid.

**Logical OR:**

- Only one precedence constraint that points to the constrained executable must evaluate to true in order for that executable to run. If this option is selected, the arrow is dotted.



Copyright © Capgemini 2015. All Rights Reserved 54

To add an expression to a precedence constraint, the first step is to select Evaluation operation. And Evaluation operation has following options:

**Constraint:** The precedence constraint is evaluated solely on the option selected in the Value property. For example, if Constraint is selected as the Evaluation operation option and Success is selected as the Value option (the default settings for both properties), the precedence constraint will evaluate to true only if the precedence executable runs successfully. When the precedence constraint evaluates to true, the workflow continues and the constrained executable runs. When the Constraint option is selected, the Expression property is greyed out.

**Expression:** The precedence constraint is evaluated based on the expression defined in the Expression text box. If the expression evaluates to true, the workflow continues and the constrained executable runs. If the expression evaluates to false, the constrained executable does not run. When the Expression option is selected, the Value property is greyed out.

**Expression and Constraint:** The precedence constraint is evaluated based on both the Value property and the expression. Both must evaluate to true for the constrained executable to run.

**Expression or Constraint:** The precedence constraint is evaluated based on either the Value property or the expression. At least one of these properties must evaluate to true for the constrained executable to run.

Define precedence constraint Workflow by Logical AND or Logical OR

Precedence Constraint Editor dialog box has following two options at the bottom :

**Logical AND:** All precedence constraints that point to the constrained executable must evaluate to true in order for that executable to run. This is the default option. If it is selected, the arrow is solid.

**Logical OR:** Only one precedence constraint that points to the constrained executable must evaluate to true in order for that executable to run. If this option is selected, the arrow is dotted.

## Variables and Expressions

Lesson 20:

## Variables

- Variables store values that a SQL Server Integration Services package and its containers, tasks, and event handlers can use at run time.
- Can use variables in Integration Services packages for the following purposes:

Updating properties of package elements at run time

Including an in-memory lookup table

Loading variables with data values and then using them to specify a search condition in a WHERE clause.

Loading a variable with an integer and then using the value to control looping within a package control flow.

Populating parameter values for Transact-SQL statements at run time.

Building expressions that include variable values.



Copyright © Capgemini 2015. All Rights Reserved 56

➤ Variables store values that a SQL Server Integration Services package and its containers, tasks, and event handlers can use at run time. The scripts in the Script task and the Script component can also use variables.

- Can use variables in Integration Services packages for the following purposes:
- Updating properties of package elements at run time. For example, can dynamically set the number of concurrent executables that a Foreach Loop container allows.
  - Including an in-memory lookup table. For example, a package can run an Execute SQL task that loads a variable with data values.
  - Loading variables with data values and then using them to specify a search condition in a WHERE clause. For example, the script in a Script task can update the value of a variable that is used by a Transact-SQL statement in an Execute SQL task.
  - Loading a variable with an integer and then using the value to control looping within a package control flow. For example, can use a variable in the evaluation expression of a For Loop container to control iteration.
  - Populating parameter values for Transact-SQL statements at run time. For example, a package can run an Execute SQL task and then use variables to dynamically set the parameters in a Transact-SQL statement.
  - Building expressions that include variable values. For example, the Derived Column transformation can populate a column with the result obtained by multiplying a variable value by a column value..

➤ **Type of variable**

➤ SSIS supports following two types of variables:

➤ **System variables:** System variables are defined by Integration Services and cannot create additional system variables. System variables contain useful information about a package, container, task, or event handler. For example, at run time the **MachineName** system variable contains the name of the computer on which the package is running and **StartTime** the time the package started to run. System variables are read-only.

➤ **User-defined variables :** User-defined variables are defined by package developers, and You can create as many user-defined variables as a package requires.

➤ The names of user-defined and system variables are case sensitive.

## Variables...

- Type of variable
- SSIS supports following two types of variables:

### System variables

- System variables are defined by Integration Services and cannot create additional system variables. System variables contain useful information about a package, container, task, or event handler.

### User-defined variables

- User-defined variables are defined by package developers, and You can create as many user-defined variables as a package requires.

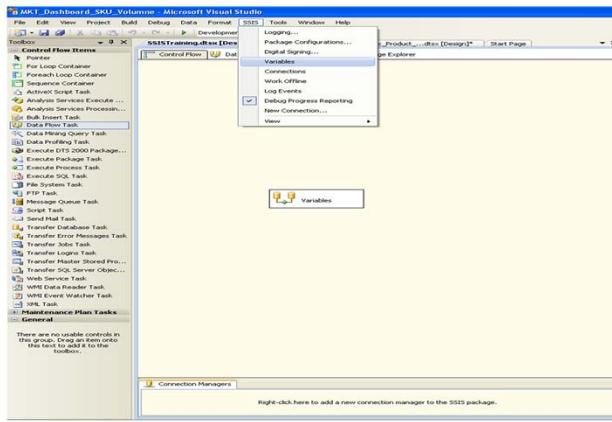
- The names of user-defined and system variables are case sensitive.



Copyright © Capgemini 2015. All Rights Reserved 57

## Variables

- To add a variable
- On the SSIS menu, click Variables or right click on control flow and select variables:



The screenshot shows the Microsoft Visual Studio interface with the title bar "AMT Dashboard SSO - Microsoft Visual Studio". The menu bar is visible with "File", "Edit", "View", "Project", "Build", "Debug", "Data", "Format", "Tools", "Windows", and "Help". A context menu is open over the "Control Flow" tab in the tabs bar, with "Variables" highlighted. The main workspace shows a blank "Control Flow" tab. At the bottom, there is a "Connection Managers" section with the text "Right-click here to add a new connection manager to the SSIS package." The Capgemini logo is at the bottom left, and the copyright information "Copyright © Capgemini 2015. All Rights Reserved 58" is at the bottom right.

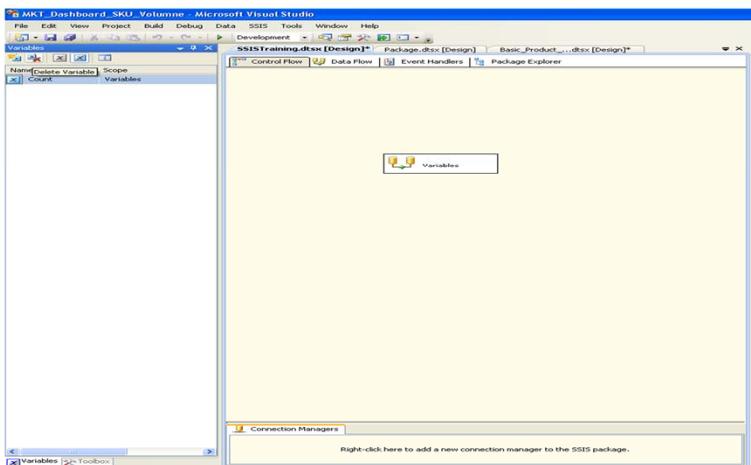
## Variables

- In the Variables window, click the Add Variable icon. The new variable is added to the list:

The screenshot shows the Microsoft Visual Studio interface for an SSIS package named 'SSISTraining.dtsx'. The 'Variables' window is open on the left, displaying a list of variables. An 'Add Variable' button is visible at the top of the list. The main design surface shows a single 'Variable' node under the 'Control Flow' tab. The 'Connection Managers' window at the bottom right is empty, with a placeholder message: 'Right-click here to add a new connection manager to the SSIS package.' The status bar at the bottom right indicates 'Copyright © Capgemini 2015. All Rights Reserved 59'.

## Variables

- To delete a variable



SSISTraining.dtsx [Design] \* Package.dtsx [Design] Basic\_Product...dtsx [Design] \*  
Control Flow Data Flow Event Handlers Package Explorer

Variables

Connection Managers

Right-click here to add a new connection manager to the SSIS package.

Capgemini CONSULTING TECHNOLOGY OUTSOURCING

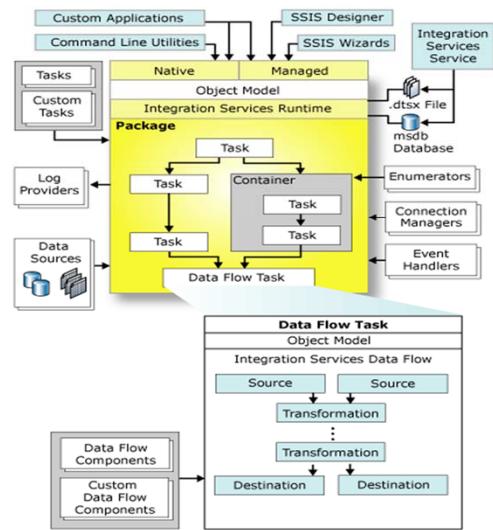
Copyright © Capgemini 2015. All Rights Reserved 60

## SSIS Architecture

Lesson 21:

## Integration Services: Architecture

- Control Flow (Runtime)
  - A parallel workflow engine
  - Executes containers and tasks
- Data Flow (“Pipeline”)
  - A special runtime task
  - A high-performance data pipeline
  - Applies graphs of components to data movement
  - Component can be sources, transformations or destinations
  - Highly parallel operations possible



## Distinguishing between data flow pipeline and package runtime

- Runtime Engine :

- The SSIS Runtime Engine manages the workflow of the packages during runtime, which means its role is to execute the tasks in a defined sequence. As you know, you can define the sequence using precedence constraints. This engine is also responsible for providing support for event logging, breakpoints in the BIDS designer, package configuration, transactions and connections. The SSIS Runtime engine has been designed to support concurrent/parallel execution of tasks in the package.

- Dataflow Pipeline Engine:

- The Dataflow Pipeline Engine is responsible for executing the data flow tasks of the package. It creates a dataflow pipeline by allocating in-memory structure for storing data in-transit. This means, the engine pulls data from source, stores it in memory, executes the required transformation in the data stored in memory and finally loads the data to the destination. Like the SSIS runtime engine, the Dataflow pipeline has been designed to do its work in parallel by creating multiple threads and enabling them to run multiple execution trees/units in parallel. Deploying packages to the SSISDB



Copyright © Capgemini 2015. All Rights Reserved 63

## Executing packages on the client side or hosted in the SSIS service

- SSIS client components are installed when you install Workstation components - that gives you BIDS (with ability to design and debug SSIS projects and packages) and SSMS (with ability to connect to SSIS service and monitor package execution).
- To run packages in production you need SSIS server components, which are installed when you install Integration Services. That requires server license.
- choose Workstation components, Books Online and development tools.
- That'll install the following components:
  - .BOL
  - .Native client
  - .BIDS
  - .SQLXML4
  - .MSXML6
  - .OWCL11 (Office Windows Components)
  - .SQL Server Backward-Compatibility Files
- Run setup. Choose the [Integration Services and Workstation components options](#).



Copyright © Capgemini 2015. All Rights Reserved 64

## Error Handling, Logging and Transactions

Lesson 22:

## Integration Services Event Handlers

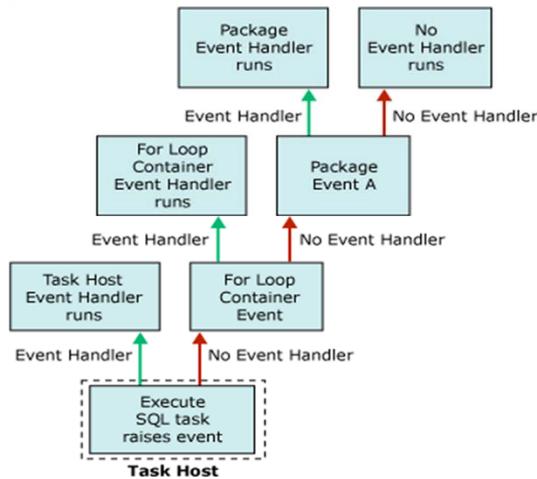
At run time, execu tables (packages and For each Loop, For Loop, Sequence, and task host containers) raise events. For example, an On Error event is raised when an error occurs. You can create custom event handlers for these events to extend package functionality and make packages easier to manage at run time. Event handlers can perform tasks such as the following:

- Clean up temporary data storage when a 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.



Copyright © Capgemini 2015. All Rights Reserved 66

## Event Handlers Hierarchy



SSIS Event handlers: The events bubble up their parent and ancestor hierarchy until they reach the package. Each ancestor, in turn, gets a chance to execute their event handler. If there is no event handler defined, the event just bubbles up to the next parent. If no event handler is defined for the event anywhere in the hierarchy, then event is lost.

## Options to disable -

- a) System::Propagate
  - b) to filter at the event handler itself  
(i.e. using System::SourceName)

We can make use of this functionality for one place logging or show progressive counts or status



Copyright © Capgemini 2015. All Rights Reserved

## Event Handler Configuration

Event handlers are members of an event handler collection, and all containers include this collection. If you create the package using SSIS Designer, you can see the members of the event handler collections in the Event Handlers folders on the Package Explorer tab of SSIS Designer.

- You can configure the event handler container in the following ways:
  - Specify a name and description for the event handler.  
Indicate whether the event handler runs, whether the package fails if the event handler fails, and the number of errors that can occur before the event handler fails.
  - Specify an execution result to return instead of the actual execution result that the event handler returns at run time.
  - Specify the transaction option for the event handler.
  - Specify the logging mode that the event handler uses.



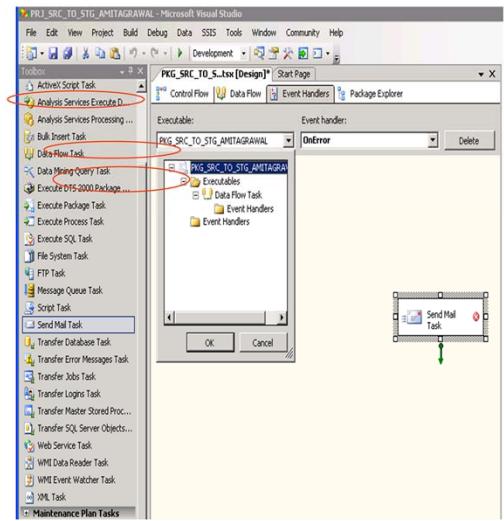
Copyright © Capgemini 2015. All Rights Reserved 68

## Event Handler Configuration

1) Execu tables: It define various execu tables in a project.

2) Event Handlers can be applied at various levels.

- a) Package level.
- b) Data Flow level.

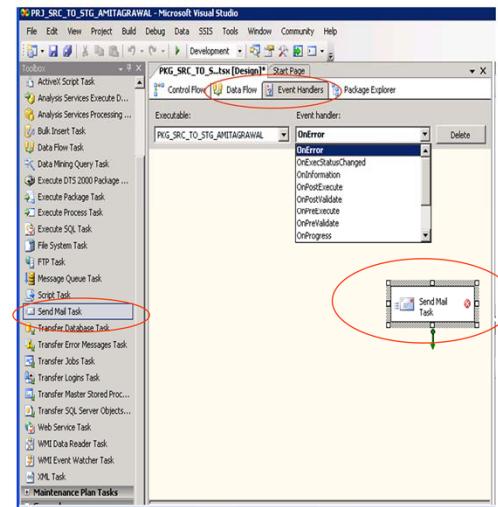


Copyright © Capgemini 2015. All Rights Reserved 69

## Event Handler Configuration

### Event Handlers:

- There is drop down list we can select various events from it.
- Toolbox: We can select and assign various task by drag – drop to a event handler.
- For eg. In this snap shot we can see there is send mail task onError.



## Run Time Events

### Event handler

#### On Error

### Event

The event handler for the On Error event. This event is raised by an executable when an error occurs.

#### On Exec Status Changed

The event handler for the On Exec Status Changed event. This event is raised by an executable when its execution status changes

#### On Information

The event handler for the On Information 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.

#### On Post Execute

The event handler for the **On Post Execute** event. This event is raised by an executable immediately after it has finished running.



Copyright © Capgemini 2015. All Rights Reserved 71

## Run Time Events

On Post Validate

The event handler for the On Post Validate event. This event is raised by an executable when its validation is finished.

On Pre Execute

The event handler for the On Pre Execute event. This event is raised by an executable immediately before it runs.

On Pre Validate

The event handler for the On Pre Validate event. This event is raised by an executable when its validation starts.

On Progress

The event handler for the On Progress event. This event is raised by an executable when measurable progress is made by the executable.



Copyright © Capgemini 2015. All Rights Reserved 72

## Run Time Events

On Query Cancel

The event handler for the On Query Cancel event. This event is raised by an executable to determine whether it should stop running.

OnTaskFailed

The event handler for the On Task Failed event. This event is raised by a task when it fails.

OnVariableValueChanged

The event handler for the On Variable Value Changed 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 Raise Change Event property for the variable to False.

OnWarning

The event handler for the On Warning event. This event is raised by an executable when a warning occurs.



Copyright © Capgemini 2015. All Rights Reserved 73

## Security

- Encryption - Using encryption to secure packages or parts of packages.
- Sensitive Data Protection - Identifying and protecting passwords and other sensitive data.
- Roles - SQL Server roles to control access to packages stored in SQL Server. Operator, Limited User, and Admin.
- Digital Signing - Code signing certificate to ensure a package hasn't changed.
- Define protection level appropriately to add security to package.



Copyright © Capgemini 2015. All Rights Reserved 74

## **Execution , debugging and Protection**

Lesson 23:

## Debugging

- Expanded Error, Warning and Information Reporting
- Watch windows
- Data Viewers in the data flow task
- Breakpoints – Script task and package breakpoints integrated
- Error, warning and Information view window
- Log event viewing in log window
- All events logged



Copyright © Capgemini 2015. All Rights Reserved 76

## Restartability - Checkpoints

- Enables recovery from failed state
- Saves work at the task level
- Variable values saved
- Saves to XML temporary file
- Checkpoint file removed when package restarts
- Runtime feature, not dataflow restartability



Copyright © Capgemini 2015. All Rights Reserved 77

# Deployment

Lesson 24:

## Difference between Project and Package Deployment Model

Project Deployment Model	Package Deployment Model
A project is the unit of deployment.	A package is the unit of deployment.
Parameters are used to assign values to package properties.	Configurations are used to assign values to package properties.
A project, containing packages and parameters, is built to a project deployment file (.ispac extension).	Packages (.dtsx extension) and configurations (.dtsConfig extension) are saved individually to the file system.
A project, containing packages and parameters, is deployed to the SSISDB catalog on an instance of SQL Server.	Packages and configurations are copied to the file system on another computer. Packages can also be saved to the MSDB database on an instance of SQL Server.



## Switching between project and package deployment modes

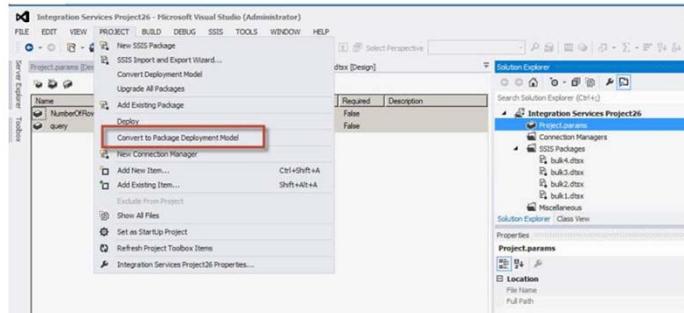
- SQL Server 2005 and 2008 use the legacy Package Deployment Model. This model is based on the package, which is the unit of deployment. However, SSIS packages usually interact with others and it is necessary to pass connection and parameters from one package to another. The new model is the Project Deployment Model which means deployment is per project and not per package.
- In SQL Server 2008, we store all the information in configuration files. Now we store the configuration information in project connections and project parameters. As per below we can switch between project and package deployment modes
  - Convert Project Deployment Model to legacy Package Deployment Model
  - Convert legacy Package Deployment Model to Project Deployment Model



Copyright © Capgemini 2015. All Rights Reserved 80

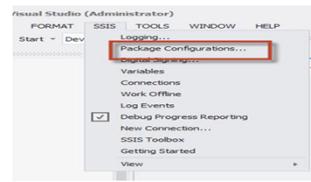
## Convert Project Deployment Model to Legacy Package Deployment Model in SSIS

- Open SSDT and create a SSIS project. In the solution parameters, add some parameters to the project.
- In order to convert this Project Deployment Model to the legacy Package Deployment Model, go to Project > Convert to Package Deployment Model:



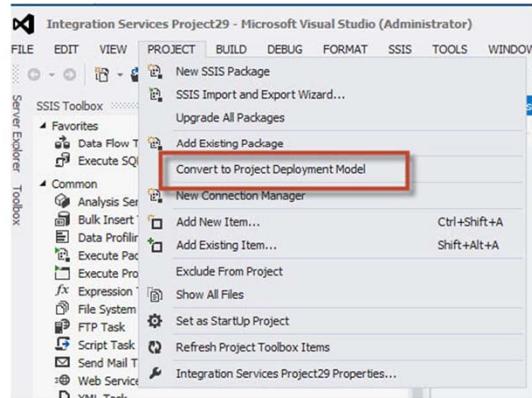
## Convert Project Deployment Model to Legacy Package Deployment Model in SSIS

- You will receive an error message related to the compatibility check. It will verify if the package can be converted:
- if you check the messages, you will notice that these new features are not compatible with the Package Deployment Model. This is because Project Parameters are a new feature not supported in the old Package Deployment Model:
- Remove the project parameters in the package and try again. If everything is OK, the menu will show the Package Configurations option. This option is only available when you convert from Project Model to Package Model.



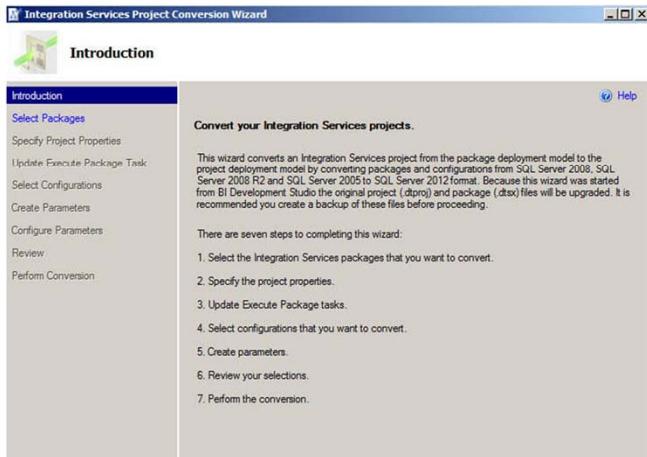
## Convert Legacy Package Deployment Model to Project Deployment Model in SSIS

- Go to Project > Convert to Project Deployment Model:



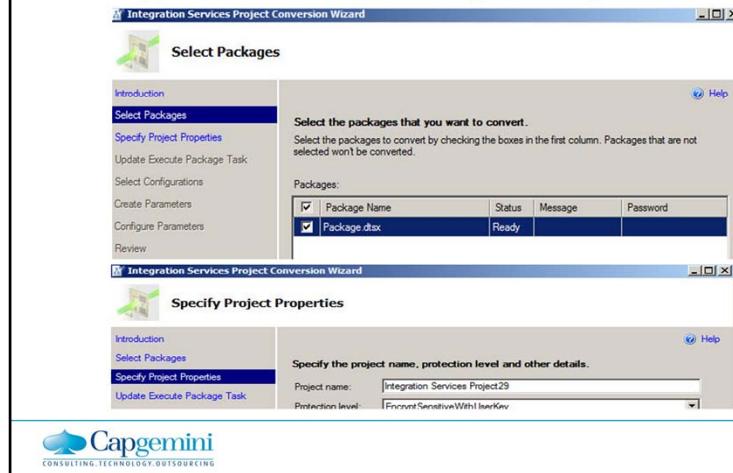
## Convert Legacy Package Deployment Model to Project Deployment Model in SSIS

- The Integration Services Project Conversion Wizard will be displayed:



## Convert Legacy Package Deployment Model to Project Deployment Model in SSIS

- Select the Package you want to convert
- Specify the project name and the protection level (by default, the EncryptSensitiveWithUserKey is used):



## Convert Legacy Package Deployment Model to Project Deployment Model in SSIS

- The next step is to update the execution Tasks:

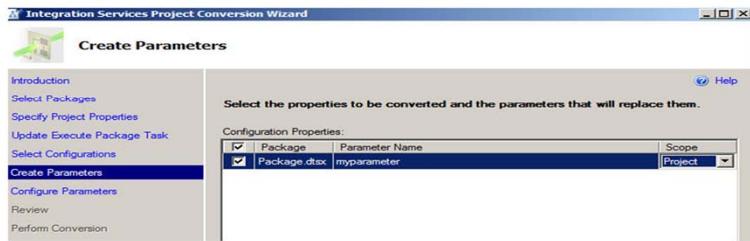


- You can select the existing configurations (the XML file that we created)



## Convert Legacy Package Deployment Model to Project Deployment Model in SSIS

- In Create Parameters, we will create a project parameter to store the information of the XML configuration file of the previous step. You can assign a name and the scope for the parameter:



- Specify and configure the values of the parameter and then can see result. Once converted will notice in the menu that the option to create **Package Configuration** disappeared.

## Running packages from SQL Server

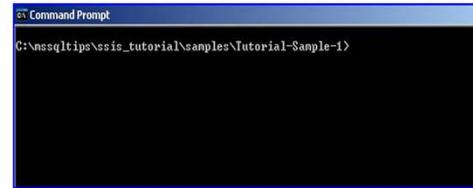
- Four options available to execute an SSIS package:
- DTEXEC command line utility
- DTEXECUI windows application
- SSISDB
- SQL Server Agent



Copyright © Capgemini 2015. All Rights Reserved 88

## Execute Package using DTEXEC

SQL Server includes the command line tool DTEXEC.EXE which can be used to execute an SSIS package. DTEXEC can be run from a Command Prompt or from a batch (.BAT) file.



Navigate to the Tutorial-Sample-1

DTEXEC /FILE CreateSalesForecastInput.dtsx

Type the following command to execute the CreateSalesForecastInput.dtsx package

DTEXEC /?

To see the complete list of command line options for DTEXEC type:



Copyright © Capgemini 2015. All Rights Reserved 89

### Overview

SQL Server includes the command line tool DTEXEC.EXE which can be used to execute an SSIS package. DTEXEC can be run from a Command Prompt or from a batch (.BAT) file.

### Explanation

To begin open a Command Prompt and navigate to the Tutorial-Sample-1 project folder as shown below:

It is not necessary to run DTEXEC from the folder where the SSIS package is located; it's just easier to change to the directory for demonstration purposes. Type the following command to execute the CreateSalesForecastInput.dtsx package:

DTEXEC /FILE CreateSalesForecastInput.dtsx

To see the complete list of command line options for DTEXEC type:  
DTEXEC /?

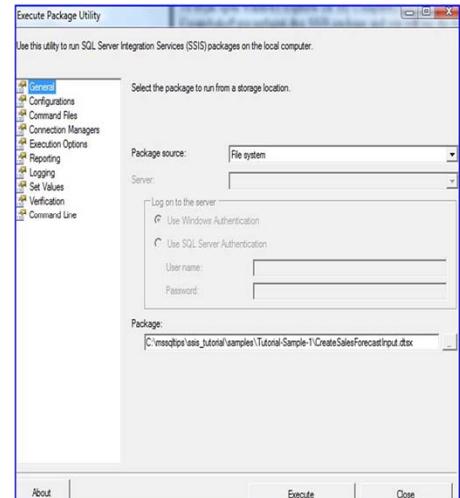
## Execute Package using DTEXECUI

Navigate to the Tutorial-Sample-1 project folder

Double-click on the SSIS package

As a general rule you can simply click the Execute button to run your package.

Fine tune your execution by clicking through the various screens and entering your own settings



Copyright © Capgemini 2015. All Rights Reserved 90

### Windows application to execute SSIS packages

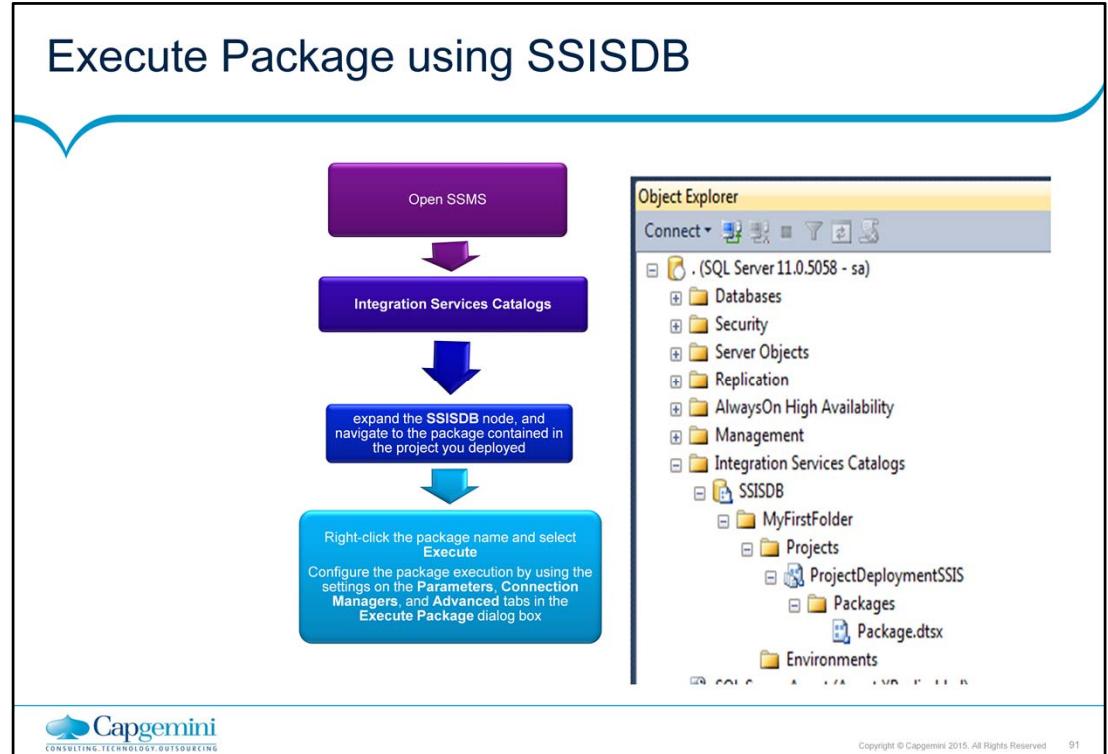
#### (DTEXECUI) Overview

SQL Server includes the Windows application DTEXECUI.EXE which can be used to execute an SSIS package. DTEXECUI provides a graphical user interface that can be used to specify the various options to be set when executing an SSIS package. You can launch DTEXECUI by double-clicking on an SSIS package file (.dtsx). You can also launch DTEXECUI from a Command Prompt then specify the package to execute.

#### Explanation

To begin open Windows Explorer (or My Computer) and navigate to the Tutorial-Sample-1 project folder. Double-click on the CreateSalesForecastInput.dtsx SSIS package and you will see the following multi-page dialog displayed:

As you can see there are many settings available when you use this utility. As a general rule you can simply click the Execute button to run your package. You can also fine tune your execution by clicking through the various screens and entering your own settings. After changing the settings click on Command Line which will show you the DTEXEC command line based on the settings you have chosen.



Open SQL Server Management Studio and connect to the instance of SQL Server that contains the Integration Services catalog.

In Object Explorer, expand the **Integration Services Catalogs** node, expand the **SSISDB** node, and navigate to the package contained in the project you deployed.

Right-click the package name and select **Execute**.

Configure the package execution by using the settings on the **Parameters**, **Connection Managers**, and **Advanced** tabs in the **Execute Package** dialog box.

Click **OK** to run the package.

-or-

Use stored procedures to run the package. Click **Script** to generate the Transact-SQL statement that creates an instance of the execution and starts an instance of the execution. The statement includes a call to the catalog.create\_execution, catalog.set\_execution\_parameter\_value, and catalog.start\_execution stored procedures. For more information about these stored procedures

## Execute Package using SQL Agent

```

graph TD
    A[Open SSMS] --> B[connect to the Database Engine]
    B --> C["drill down to the SQL Server Agent node in the Object Explorer  
Right click on the Jobs node and select New Job"]
    C --> D["Go to the Steps page, click New,  
and fill in the dialog as shown"]
  
```

The flowchart illustrates the process of executing an SSIS package using SQL Agent. It starts with opening SSMS, connecting to the database engine, drilling down to the SQL Server Agent node in the Object Explorer, right-clicking on the Jobs node, and selecting 'New Job'. Finally, it shows navigating to the Steps page, clicking 'New', and filling in the dialog as shown in the screenshot.

The screenshot shows the 'Create Sales Forecast Input Spreadsheet' dialog in SQL Server Agent. The 'Step name' is 'Create Sales Forecast Input Spreadsheet', 'Type' is 'SQL Server Integration Services Package', and 'Run as' is 'SQL Agent Service Account'. The 'Package source' is set to 'SQL Server' and the 'Server' is 'VSSQLDB02'. Under 'Log on to the server', 'Use Windows Authentication' is selected. The 'Package' field contains '\\CreateSalesForecastInput'. Buttons for 'OK', 'Cancel', 'Next', and 'Previous' are visible at the bottom.

**Capgemini**  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 92

### (SQL Server Agent) Overview

SQL Server Agent includes the SQL Server Integration Services Package job step type which allows you to execute an SSIS package in a SQL Server Agent job step. This can be especially handy as it allows you to schedule the execution of an SSIS package so that it runs without any user interaction.

### Explanation

To begin open SSMS, connect to the Database Engine, and drill down to the SQL Server Agent node in the Object Explorer. Right click on the Jobs node and select New Job from the popup menu. Go to the Steps page, click New, and fill in the dialog as shown below:

In the example above the SSIS package to be executed is deployed to SQL Server (i.e. the MSDB database). You can also execute packages deployed to the file system or the SSIS package store.

Note that the Run as setting is the SQL Agent Service Account. This is the default setting although from a security standpoint it may not be what you want. You can setup a Proxy that allows you to give a particular credential permission to execute an SSIS package from a SQL Server Agent job step.

The first step to setting up the proxy is to create a credential (alternatively you could use an existing credential). Navigate to Security then Credentials in SSMS Object Explorer and right click to create a new credential as shown below:

Navigate to SQL Server Agent then Proxies in SSMS Object Explorer and right click to create a new proxy as shown below:

You must specify the credential and check SQL Server Integration Services Package. Now when you create or edit a SQL Server Agent job step, you can specify the proxy for the Run as setting as shown below:

## Scheduling .....

Set up the proxy is to create a credentials

Navigate to Security then Credentials

Right click to create a new credential as shown

Navigate to SQL Server Agent then Proxies in SSMS Object Explorer

Right click to create a new proxy as shown

The screenshot shows the 'Proxies' node under 'SQL Server Agent' in the Object Explorer. A new proxy is being created with the following details:

- Proxy name:** SSIS Proxy
- Credential name:** SSIS Execution Account
- Description:** (empty)

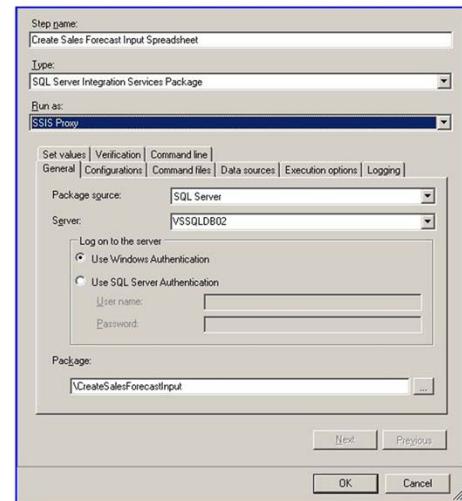
Under the 'Active to the following subsystems:' section, the checkbox for 'SQL Server Integration Services Package' is checked.

Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 93

## Scheduling .....

Specify the credential and check SQL Server Integration Services Package. Now when you create or edit a SQL Server Agent job step, you can specify the proxy for the Run as setting as shown



## Troubleshoot a SSIS package that doesn't run when you call the package from a SQL Server Agent job step

- The recommended methods for resolving this issue, including creating a proxy account, modifying the package Protection Level property setting, saving sensitive data in a package configuration file, and storing a package in the SQL Server msdb database.
- When you call a package from a SQL Server Agent job step and the package doesn't run, one of the following conditions is true:
  - The user account that runs the package as a job step differs from the original package author.

—or—
  - The user account does not have the required permissions to make connections or to access resources outside the package
  -



Copyright © Capgemini 2015. All Rights Reserved 95

The recommended methods for resolving this issue, including creating a proxy account, modifying the package ProtectionLevel property setting, saving sensitive data in a package configuration file, and storing a package in the SQL Server msdb database. When you call a package from a SQL Server Agent job step and the package doesn't run, one of the following conditions is true:

The user account that runs the package as a job step differs from the original package author.

—or—

The user account does not have the required permissions to make connections or to access resources outside the package

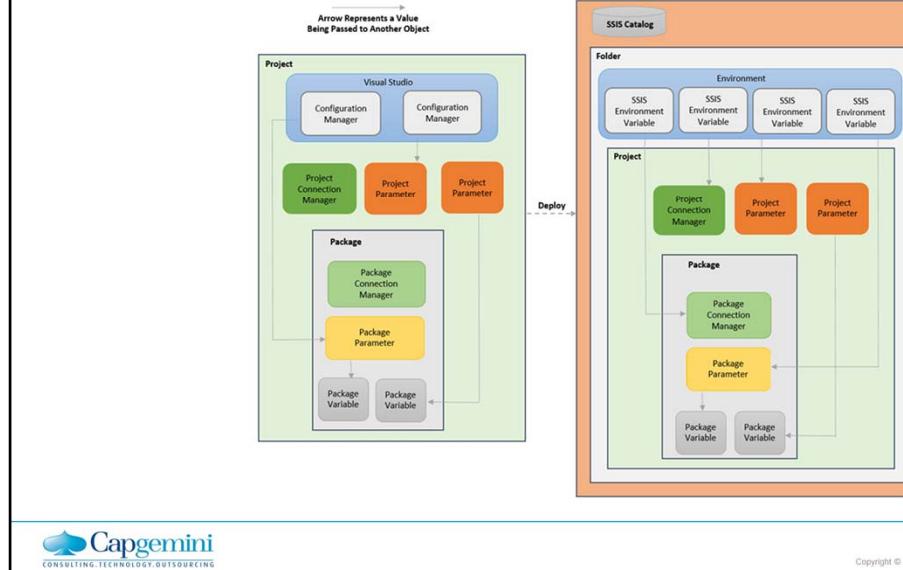
When the user account that calls the package from the job step differs from the original package author, the package protection level may prevent the package from running.

This is because the user account either cannot decrypt the package or the package-sensitive data, or the user account cannot provide the sensitive data that is missing from the package.

Examples of sensitive data are the password part of a connection string, a variable that is marked sensitive, etc.

There are a couple of recommended methods for resolving issues with encryption and sensitive data.

## Leveraging Project/package parameters



## SSIS Project & Package Parameters

- Project Parameters
- Project parameters are new with SSIS 2012. A project parameter can be shared among all of the packages in an SSIS project. You want to use a project parameter when, at run-time, the value is the same for all packages.

Name	Data type	Value	Sensitive	Required	Description
FlatFileFolder_Subsiday1	String	\\\wRNSV2BI_FlatFiles\Subsiday1	False	False	
FlatFileFolder_Subsiday2	String	\\\wRNSV2BI_FlatFiles\Subsiday2	False	False	

Project Parameter

**Package Parameters**

Package parameters are also new with SSIS 2012. A package parameter is exactly the same as a project parameter – except that the scope of a package parameter is the individual package it resides in. You want to use a package parameter when, at run-time, the value is different for each package.

Name	Data type	Value	Sensitive	Required	Description
SummaryOrDetailValuesExpected	String	Summary	False	False	

Package Parameter



Copyright © Capgemini 2015. All Rights Reserved 97

## Use of package and project parameters

- Parameters are useful for providing runtime values to properties during a package execution. Hence, in a way they replace the concept of configurations that we had for SSIS packages.
- Package parameters are useful for providing values to specific package executions. As they are package scope, they would only be available to the package in which they were created.
- Project parameters are available to all the packages in a project. They are useful for configuring values which can be shared between packages. For instance if you wanted to have a single parameter containing a server name to be used by multiple packages, then Project Parameters will be useful for you.
- You can use a parameter anywhere in an expression by referring it using the following syntax:
- `[$<<Project/Package>>:<<Parameter Name>>]`
- The evaluation order is similar to using a variable in an expression and then executing a package. Hence, the value gets evaluated and assigned to the property at Validation phase.



Copyright © Capgemini 2015. All Rights Reserved 98

# Performance Tuning

Lesson 25:

## Integration Services: Internals

- Two Engines:

- Runtime engine {highly parallel control flow engine that coordinates the execution of tasks or units of work within SSIS and manages the engine threads that carry out those tasks.}
- Data flow “pipeline” engine {manages the data pipeline}

- Inside the Data Flow

- Logical pipeline
- Physical memory buffers
- Execution Trees {is a group of transformations which start at either a source adapter or an asynchronous transform and end at the first asynchronous transform or a destination adapter. A data buffer has the scope of an execution tree.}
- Changes in SSIS 2008: Execution Trees are referred as “paths”, that a path can be further divided into “subpaths.”



Copyright © Capgemini 2015. All Rights Reserved 100

## Integration Services: Internals

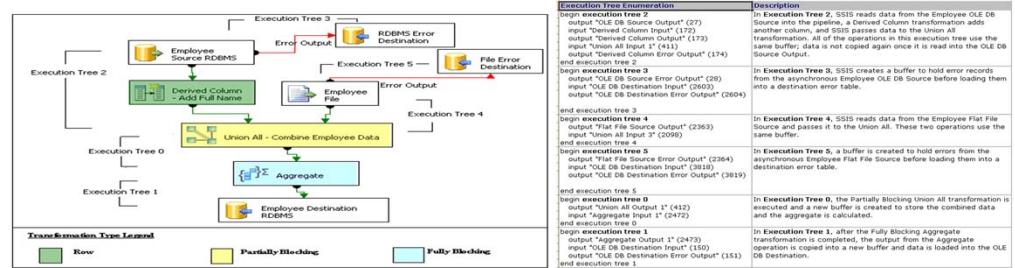
- Row Transformation (synchronous transformation):
  - Row-by-row basis
  - Do not block data flow in the pipeline
  - Data is not copied around, only pointers
  - Examples: Data Conversion, Derived Columns, Copy column, Multicast, Row count, Lookup etc.
- Partially Blocking Transformation (asynchronous transformation):
  - Introduces new buffers in memory layout
  - Transformed data is copied into new buffers
  - Examples: Merge, Merge Join, Union All etc.
- Blocking Transformation (asynchronous transformation):
  - Must see all data before passing on rows
  - Block the data flow – can be heavy on memory
  - May also use “private buffers” to assist with transforming data
  - Examples: Sort, Aggregate etc.



Copyright © Capgemini 2015. All Rights Reserved 101

## Integration Services: Internals (Execution Trees)

- Execution trees help us understand the buffer usage and check other alternatives for execution in terms of improving performance.{See how SSIS creates buffer for Source was observed and the same buffer being used by Derived Column transformation as it is a row/synchronous transformation. Separate sets of buffers are being created for Union All and Aggregate transformations as they are Partially Blocking and Blocking transformations respectively and need additional buffers to store their outcome.}



## Tuning Tips

- Remove redundant columns
  - Use SELECT statements as opposed to tables
  - SELECT \* is your enemy
  - Also remove redundant columns after every async component!
- Filter rows
  - WHERE clause is your friend
  - Conditional Split in SSIS
  - Concatenate or re-route unneeded columns
- Parallel loading
  - Source system split source data into multiple chunks
    - Flat Files – multiple files
    - Relational – via key fields and indexes- Multiple Destination components all loading same table
- Which component?
  - Bulk Import Task vs. Data Flow
    - Bulk Import might give better performance if there are no transformations or filtering required, and the destination is SQL Server.
  - Lookup vs. MergeJoin (LeftJoin) vs. set based statements in SQL
    - MergeJoin might be required if you're not able to populate the lookup cache.
    - Set based SQL statements might provide a way to persist lookup cache misses and apply a set based operation for higher performance.
  - Script vs. custom component: Script might be good enough for small transforms that're typically not reused



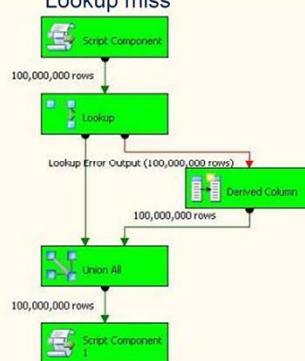
SQL Server 2005 SSIS Tuning the Dataflow Task.htm



Copyright © Capgemini 2015. All Rights Reserved 103

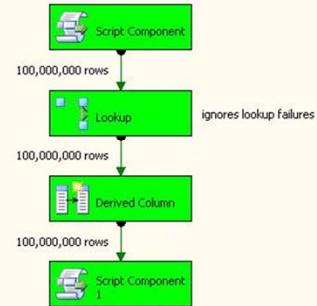
## Performance Tuning: Case Study

Use Error Output for handling Lookup miss



105 seconds

Ignore lookup errors and check for null looked up values in Derived Column



83 seconds



Copyright © Capgemini 2015. All Rights Reserved 104

## Best Practices

Lesson 26:

## Best Practices : Learning from Previous Projects

- Use template packages whenever possible. This enables consistency in
  - Creating Variables
  - Connection Manager
  - Setting common package properties
  - Using same Config Files across packages
- Use OLE DB connections unless there is a compelling reason to do otherwise.
- Only Configure package variables.
- One target table per package.
- Write comments ("annotate") in the designer editor, describing the functionality of the task, transformation etc.
- For large volume data, avoid row based operations, instead use Update set operations.
- Minimize blocking: As far as possible avoid use of asynchronous transformation.
- Maintain separate development, UAT and production environment. Also use source control tool (like VSS) – to check-in package .xml or solution files.
- BufferTempStoragePath - The location for temporary storage of buffer data. In case of space issues on the primary drive [in most of cases the C:/], this property should be set to the alternate path.



Copyright © Capgemini 2015. All Rights Reserved 106

## Best Practices : Learning from Previous Projects

- If one package is being called from other package, the called package can execute either in its own memory or in the memory of the parent package. Setting the Execute Out Of Process property of the Execute Package Task to true. This forces SSIS to create a separate process for each child package to run in
- Explicitly Type Casting in Derive column Transformation increases the overhead. Usage of Data conversion Transformation to do the necessary casting will resolve the problem
- Ignoring Execution Status of the Dataflow task, set 'Force Execution Result' property of current task to 'Completion'.
- Limit the number of data flows in a single package. To avoid memory issues, while opening, saving and executing SSIS packages. Break the package logically into multiple packages.
- Observation: Package stored on file system executes faster than ones stored on database. Also debugging on file system is better.
- Maximize Parallelism: caution-Multiple parallel flows at time slows down the process depending on the available server resources-Some properties that impact parallel runs are
  - Max Concurrent Executables (Package property)- define how many tasks can run concurrently,
  - Engine Threads (Data Flow property) - how threads the engine would create
  - Data Flow Performance Improvement, properties that impact - 'DefaultBufferSize' to 100000000. 'DefaultBufferMaxRows' to 500000 {depending on server resources}.
  - Loading tables with no constraints. Use - FAST Load option, this is similar to bulk insert operation, which generally can be used for staging table loads. The performance will be faster than the normal load



Copyright © Capgemini 2015. All Rights Reserved 107