SSIS-IQ

Question 1 - True or False - Using a checkpoint file in SSIS is just like issuing the CHECKPOINT command against the relational engine. It commits all of the data to the database.

False. SSIS provides a Checkpoint capability which allows a package to restart at the point of failure.

Question 2 - Can you explain the what the Import\Export tool does and the basic steps in the wizard?

The Import\Export tool is accessible via BIDS or executing the dtswizard command.
The tool identifies a data source and a destination to move data either within 1 database, between instances or even from a database to a file (or vice versa).

Question 3 - What are the command line tools to execute SQL Server Integration Services packages?

DTSEXECUI - When this command line tool is run a user interface is loaded in order to configure each of the applicable parameters to execute an SSIS package.
DTEXEC - This is a pure command line tool where all of the needed switches must be passed into the command for successful execution of the SSIS package.

Question 4 - Can you explain the SQL Server Integration Services functionality in Management Studio?

You have the ability to do the following:
Login to the SQL Server Integration Services instance
View the SSIS log
View the packages that are currently running on that instance
Browse the packages stored in MSDB or the file system
Import or export packages
Delete packages
Run packages

Question 5 - Can you name some of the core SSIS components in the Business Intelligence Development Studio you work with on a regular basis when building an SSIS package?

Connection Managers
Control Flow
Data Flow
Event Handlers
Variables window
Toolbox window
Output window
Logging
Package Configurations

Question Difficulty = Moderate

Question 1 - True or False: SSIS has a default means to log all records updated, deleted or inserted on a per table basis.

False, but a custom solution can be built to meet these needs.

Question 2 - What is a breakpoint in SSIS? How is it setup? How do you disable it?

A breakpoint is a stopping point in the code. The breakpoint can give the Developer\DBA an opportunity to review the status of the data, variables and the overall status of the SSIS package. 10 unique conditions exist for each breakpoint.
Breakpoints are setup in BIDS. In BIDS, navigate to the control flow interface. Right click on the object where you want to set the breakpoint and select the 'Edit Breakpoints...' option.

Question 3 - Can you name 5 or more of the native SSIS connection managers?

OLEDB connection - Used to connect to any data source requiring an OLEDB connection (i.e., SQL Server 2000)
Flat file connection - Used to make a connection to a single file in the File System. Required for reading information from a File System flat file
ADO.Net connection - Uses the .Net Provider to make a connection to SQL Server 2005 or other connection exposed through managed code (like C#) in a custom task
Analysis Services connection - Used to make a connection to an Analysis Services database or project. Required for the Analysis Services DDL Task and Analysis Services Processing Task
File connection - Used to reference a file or folder. The options are to either use or create a file or folder
Excel
FTP
HTTP
MSMQ
SMO
SMTP
SQLMobile
WMI

Question 4 - How do you eliminate quotes from being uploaded from a flat file to SQL Server?

In the SSIS package on the Flat File Connection Manager Editor, enter quotes into the Text qualifier field then preview the data to ensure the quotes are not included.
Additional information: How to strip out double quotes from an import file in SQL Server Integration Services

Question 5 - Can you name 5 or more of the main SSIS tool box widgets and their functionality?

For Loop Container
Foreach Loop Container
Sequence Container
ActiveX Script Task
Analysis Services Execute DDL Task
Analysis Services Processing Task
Bulk Insert Task

Data Flow Task
Data Mining Query Task
Execute DTS 2000 Package Task
Execute Package Task
Execute Process Task
Execute SQL Task
etc.

Question Difficulty = Difficult

Question 1 - Can you explain one approach to deploy an SSIS package?

One option is to build a deployment manifest file in BIDS, then copy the directory to the applicable SQL Server then work through the steps of the package installation wizard
A second option is using the dtutil utility to copy, paste, rename, delete an SSIS Package
A third option is to login to SQL Server Integration Services via SQL Server Management Studio then navigate to the 'Stored Packages' folder then right click on the one of the children folders or an SSIS package to access the 'Import Packages...' or 'Export Packages...'option.
A fourth option in BIDS is to navigate to File | Save Copy of Package and complete the interface.

Question 2 - Can you explain how to setup a checkpoint file in SSIS?

The following items need to be configured on the properties tab for SSIS package:
CheckpointFileName - Specify the full path to the Checkpoint file that the package uses to save the value of package variables and log completed tasks. Rather than using a hard-coded path as shown above, it's a good idea to use an expression that concatenates a path defined in a package variable and the package name.
CheckpointUsage - Determines if/how checkpoints are used. Choose from these options: Never (default), IfExists, or Always. Never indicates that you are not using Checkpoints. IfExists is the typical setting and implements the restart at the point of failure behavior. If a Checkpoint file is found it is used to restore package variable values and restart at the point of failure. If a Checkpoint file is not found the package starts execution with the first task. The Always choice raises an error if the Checkpoint file does not exist.
SaveCheckpoints - Choose from these options: True or False (default). You must select True to implement the Checkpoint behavior.

Question 3 - Can you explain different options for dynamic configurations in SSIS?

Use an XML file
Use custom variables
Use a database per environment with the variables
Use a centralized database with all variables

Question 4 - How do you upgrade an SSIS Package?

Depending on the complexity of the package, one or two techniques are typically used:
Recode the package based on the functionality in SQL Server DTS
Use the Migrate DTS 2000 Package wizard in BIDS then recode any portion of the package that is not accurate

Question 5 - Can you name five of the Perfmon counters for SSIS and the value they provide?

SQLServer:SSIS Service
SSIS Package Instances - Total number of simultaneous SSIS Packages running
SQLServer:SSIS Pipeline
BLOB bytes read - Total bytes read from binary large objects during the monitoring period.
BLOB bytes written - Total bytes written to binary large objects during the monitoring period.
BLOB files in use - Number of binary large objects files used during the data flow task during the monitoring period.
Buffer memory - The amount of physical or virtual memory used by the data flow task during the monitoring period.
Buffers in use - The number of buffers in use during the data flow task during the monitoring period.
Buffers spooled - The number of buffers written to disk during the data flow task during the monitoring period.
Flat buffer memory - The total number of blocks of memory in use by the data flow task during the monitoring period.
Flat buffers in use - The number of blocks of memory in use by the data flow task at a point in time.
Private buffer memory - The total amount of physical or virtual memory used by data transformation tasks in the data flow engine during the monitoring period.
Private buffers in use - The number of blocks of memory in use by the transformations in the data flow task at a point in time.
Rows read - Total number of input rows in use by the data flow task at a point in time.
Rows written - Total number of output rows in use by the data flow task at a point in time.
Source:
http://www.dotnetspider.com/forum/158771-Sql-Server-Integration-services-Interview-questions.aspx

---

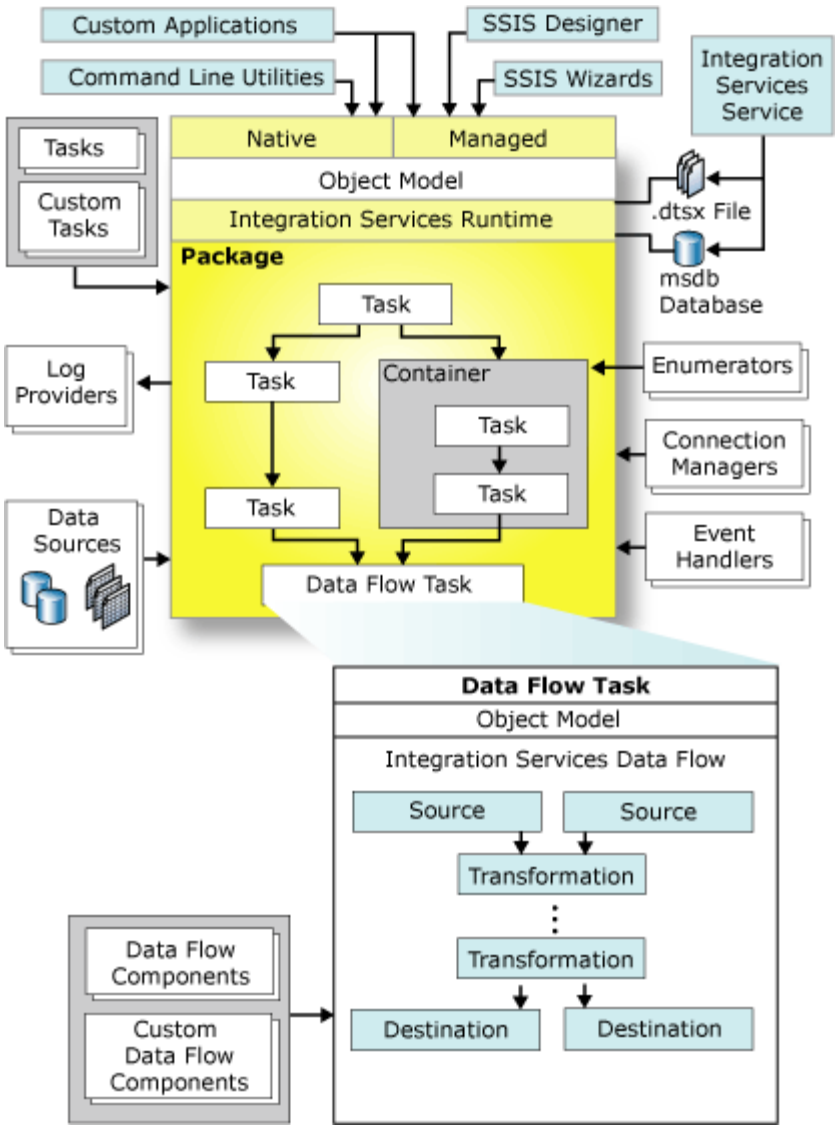
Common search for new SSIS programmer looking for change is what questions to expect on SSIS. Based on the interviews I take on SSIS, I will list down my favorites and expected questions on SSIS.

Q1 Explain architecture of SSIS?

Integration Services Architecture

Microsoft SQL Server 2005 Integration Services (SSIS) consists of four key parts: the Integration Services service, the Integration Services object model, the Integration Services runtime and the run-time executables, and the Data Flow task that encapsulates the data flow engine and the data flow components.

The following diagram shows the relationship of the parts.



Developers who access the Integration Services object model from custom clients or write custom tasks or transformations can write code by using any common language runtime (CLR) compliant language. For more information, see Integration Services Programming.

Integration Services Service

The Integration Services service, available in SQL Server Management Studio, monitors running Integration Services packages and manages the storage of packages.

For more information, click one of the following topics:

Integration Services Service

Introducing SQL Server Management Studio

Integration Services Object Model

The Integration Services object model includes managed application programming interfaces (API) for accessing Integration Services tools, command-line utilities, and custom applications.

For more information, click one of the following topics:

Integration Services Programming

Integration Services Tools and Utilities

Integration Services Runtime

The Integration Services runtime saves the layout of packages, runs packages, and provides support for logging, breakpoints, configuration, connections, and transactions. The Integration Services run-time executables are the package, containers, tasks, and event handlers that Integration Services includes, and custom tasks.

For more information, click one of the following topics:

Integration Services Packages

Integration Services Containers

Integration Services Tasks

Integration Services Event Handlers

Microsoft.SqlServer.Dts.Runtime

Integration Services Data Flow

The Data Flow task encapsulates the data flow engine. The data flow engine provides the in-memory buffers that move data from source to destination, and calls the sources that extract data from files and relational databases. The data flow engine also manages the transformations that modify data, and the destinations that load data or make data available to other processes. Integration Services data flow components are the sources, transformations, and destinations that Integration Services includes. You can also include custom components in a data flow.

For more information, click one of the following topics:

Data Flow Task

Data Flow Elements

Microsoft.SqlServer.Dts.Pipeline.Wrapper

Source: http://technet.microsoft.com/en-us/library/ms141709(SQL.90).aspx

Q2 Difference between Control Flow and Data Flow?
Very easy.

Q3 How would you do Logging in SSIS?

Log using the logging configuration inbuilt in SSIS or use Custom logging through Event handlers.

Monitoring How-to Topics (Integration Services)

This section contains procedures for adding log providers to a package and configuring logging by using the SQL Server Integration Services tools that Business Intelligence Development Studio provides.

How to: Enable Logging in a Package

How to: Enable Logging in a Package

This procedure describes how to add logs to a package, configure package-level logging, and save the logging configuration to an XML file. You can add logs only at the package level, but the package does not have to perform logging to enable logging in the containers that the package includes.

By default, the containers in the package use the same logging configuration as their parent container. For information about setting logging options for individual containers, see How to: Configure Logging by Using a Saved Configuration File.

To enable logging in a package

1. In Business Intelligence Development Studio, open the Integration Services project that contains the package you want.
2. On the **SSIS** menu, click **Logging**.
3. Select a log provider in the **Provider type** list, and then click **Add**.
4. In the **Configuration** column, select a connection manager or click **<New connection>** to create a new connection manager of the appropriate type for the log provider. Depending on the selected provider, use one of the following connection managers:
   - For Text files, use a File connection manager. For more information, see File Connection Manager
   - For SQL Server Profiler, use a File connection manager.
   - For SQL Server, use an OLE DB connection manager. For more information, see OLE DB Connection Manager.
   - For Windows Event Log, do nothing. SSIS automatically creates the log.
   - For XML files, use a File connection manager.
5. Repeat steps 3 and 4 for each log to use in the package.

   Note:

   A package can use more than one log of each type.
6. Optionally, select the package-level check box, select the logs to use for package-level logging, and then click the **Details** tab.
7. On the **Details** tab, select **Events** to log all log entries, or clear **Events** to select individual events.

8. Optionally, click **Advanced** to specify which information to log.

   Note:

   By default, all information is logged.
9. On the **Details** tab, click **Save.** The **Save As** dialog box appears. Locate the folder in which to save the logging configuration, type a file name for the new log configuration, and then click **Save**.
10. Click **OK**.
11. To save the updated package, click **Save Selected Items** on the **File** menu.

How to: Configure Logging by Using a Saved Configuration File

How to: Configure Logging by Using a Saved Configuration File

This procedure describes how to configure logging for new containers in a package by loading a previously saved logging configuration file.

By default, all containers in a package use the same logging configuration as their parent container. For example, the tasks in a Foreach Loop use the same logging configuration as the Foreach Loop.

To configure logging for a container

1. In Business Intelligence Development Studio, open the Integration Services project that contains the package you want.
2. On the **SSIS** menu, click **Logging**.
3. Expand the package tree view and select the container to configure.
4. On the **Providers and Logs** tab, select the logs to use for the container.

   Note:

   You can create logs only at the package level. For more information, see How to: Enable Logging in a Package.
5. Click the **Details** tab and click **Load**.
6. Locate the logging configuration file you want to use and click **Open**.
7. Optionally, select a different log entry to log by selecting its check box in the **Events** column. Click **Advanced** to select the type of information to log for this entry.

   Note:

   The new container may include additional log entries that are not available for the container originally used to create the logging configuration. These additional log entries must be selected manually if you want them to be logged.
8. To save the updated version of the logging configuration, click **Save**.
9. To save the updated package, click **Save Selected Items** on the **File** menu.

Source: http://msdn.microsoft.com/en-us/library/ms141710.aspx

How to: View Log Entries in the Log Events Window

How to: View Log Entries in the Log Events Window

This procedure describes how to run a package and view the log entries it writes. You can view the log entries in real time. The log entries that are written to the **Log Events** window can also be copied and saved for further analysis.

It is not necessary to write the log entries to a log to write the entries to the **Log Events** window.

To view log entries

1. In Business Intelligence Development Studio, open the Integration Services project that contains the package you want.
2. On the **SSIS** menu, click **Log Events**. You can optionally display the **Log Events** window by mapping the View.LogEvents command to a key combination of your choosing on the **Keyboard** page of the **Options** dialog box.
3. On the **Debug** menu, click **Start Debugging**.

   As the runtime encounters the events and custom messages that are enabled for logging, log entries for each event or message are written to the **Log Events** window.

4. On the **Debug** menu, click **Stop Debugging**.

   The log entries remain available in the **Log Events** window until you rerun the package, run a different package, or close Business Intelligence Development Studio.

5. View the log entries in the **Log Events** window.
6. Optionally, click the log entries to copy, right-click, and then click **Copy**.
7. Optionally, double-click a log entry, and in the **Log Entry** dialog box, view the details for a single log entry.
8. In the **Log Entry** dialog box, click the up and down arrows to display the previous or next log entry, and click the copy icon to copy the log entry.
9. Open a text editor, paste, and then save the log entry to a text file.

Source:
http://msdn.microsoft.com/en-us/library/ms141727.aspx

Q4 How would you do Error Handling?
its for you.

Q5 How to pass property value at Run time? How do you implement Package Configuration?
Package Configurations

SQL Server Integration Services provides package configurations that you can use to update the values of properties at run time. A configuration is a property/value pair that you add to a completed package. Typically, you create a package set properties on the package objects during package development, and then add the configuration to the package. When the package runs, it gets the new values of the property from the configuration. For example, by using a configuration, you can change the connection string of a connection manager, or update the value of a variable.

Package configurations provide the following benefits:

- Configurations make it easier to move packages from a development environment to a production environment. For example, a configuration can update the path of a source file, or change the name of a database or server.
- Configurations are useful when you deploy packages to many different servers. For example, a variable in the configuration for each deployed package can contain a different disk space value, and if the available disk space does not meet this value, the package does not run.
- Configurations make packages more flexible. For example, a configuration can update the value of a variable that is used in a property expression.

Integration Services supports several different methods of storing package configurations, such as XML files, tables in a SQL Server database, and environment and package variables.

Each configuration is a property/value pair. The XML configuration file and SQL Server configuration types can include multiple configurations.

The configurations are included when you create a package deployment utility for installing packages. When you install the packages, the configurations can be updated as a step in the package installation.

Note:

To become better acquainted with the concepts explained in this section, see Tutorial: Deploying Packages and Lesson 3: Adding Package Configurations of Tutorial: Creating a Simple ETL Package.

Package Configuration Types

The following table describes the package configuration types.

Type	Description
XML configuration file	An XML file contains the configurations. The XML file can include multiple configurations.
Environment variable	An environment variable contains the configuration.
Registry entry	A registry entry contains the configuration.
Parent package variable	A variable in the package contains the configuration. This configuration type is typically used to update properties in child packages.
SQL Server table	A table in a SQL Server database contains the configuration. The table can include multiple configurations.

XML Configuration Files

If you select the **XML configuration file** configuration type, you can create a new configuration file, reuse an existing file and add new configurations, or reuse an existing file but overwrite existing file content.

An XML configuration file includes two sections:

- A heading that contains information about the configuration file. This element includes attributes such as when the file was created and the name of the person who generated the file.
- Configuration elements that contain information about each configuration. This element includes attributes such as the property path and the configured value of a property.

The following XML code demonstrates the syntax of an XML configuration file. This example shows a configuration for the **Value** property of an integer variable named MyVar.

Copy Code

```
<?xml version="1.0"?>
<DTSConfiguration>
  <DTSConfigurationHeading>
    <DTSConfigurationFileInfo
      GeneratedBy="DomainName\UserName"
      GeneratedFromPackageName="Package"
      GeneratedFromPackageID="{2AF06766-817A-4E28-9878-0DE37A150648}"
      GeneratedDate="2/01/2005 5:58:09 PM"/>
  </DTSConfigurationHeading>
  <Configuration ConfiguredType="Property" Path="\Package.Variables[User::MyVar].Value"
ValueType="Int32">
    <ConfiguredValue>0</ConfiguredValue>
  </Configuration>
</DTSConfiguration>
```

Registry Entry

If you want to use a registry entry to store the configuration, you can either use an existing key or create a new key in HKEY_CURRENT_USER. The registry key that you use must have a value named **Value**. The value can be a DWORD or a string.

If you select the **Registry entry** configuration type, you type the name of the registry key in the Registry entry box. The format is <registry key>. If you want to use a registry key that is not at the root of HKEY_CURRENT_USER, use the format <registry key\registry key\...> to identify the key. For example, to use the MyPackage key located in SSISPackages, type **SSISPackages\MyPackage**.

SQL Server

If you select the **SQL Server** configuration type, you specify the connection to the SQL Server database in which you want to store the configurations. You can save the configurations to an existing table or create a new table in the specified database.

The following SQL statement shows the default CREATE TABLE statement that the Package Configuration Wizard provides.

Copy Code

```
CREATE TABLE [dbo].[SSIS Configurations]
(
ConfigurationFilter NVARCHAR(255) NOT NULL,
ConfiguredValue NVARCHAR(255) NULL,
```

PackagePath NVARCHAR(255) NOT NULL,
ConfiguredValueType NVARCHAR(20) NOT NULL
)

The name that you provide for the configuration is the value stored in the **ConfigurationFilter** column.

Direct and Indirect Configurations

Integration Services provides direct and indirect configurations. If you specify configurations directly, Integration Services creates a direct link between the configuration item and the package object property. Direct configurations are a better choice when the location of the source does not change. For example, if you are sure that all deployments in the package use the same file path, you can specify an XML configuration file.

Indirect configurations use environment variables. Instead of specifying the configuration setting directly, the configuration points to an environment variable, which in turn contains the configuration value. Using indirect configurations is a better choice when the location of the configuration can change for each deployment of a package.

http://msdn.microsoft.com/en-us/library/ms141682.aspx

Q6 How would you deploy a SSIS Package on production?
1. Create deployment utility by setting its property as true .
2. It will be created in the bin folder of the solution as soon as package is build.
3. Copy all the files in the utility and use manifest file to deply it on the Prod.

Q7 Difference between DTS and SSIS?
Every thing except both are product of Microsoft :-)

Q8 What are new features in SSIS 2008?
http://sqlserversolutions.blogspot.com/2009/01/new-improvementfeatures-in-ssis-2008.html

Q9 How would you pass a variable value to Child Package?

http://sqlserversolutions.blogspot.com/2009/02/passing-variable-to-child-package-from.html

How to: Use Values of Parent Variables in Child Packages
New: **5 December 2005**

This procedure describes how to create a package configuration that uses the parent variable configuration type to enable a child package that is run from a parent package to access a variable in the parent.

It is not necessary to create the variable in the parent package before you create the package configuration in the child package. You can add the variable to the parent package at any time, but you must use the exact name of the parent variable in the package configuration. However, before you can create a parent variable configuration, there must be an existing variable in the child package that the configuration can update. For more information about adding and configuring variables, see How to: Add a Variable to a Package Using the Variables Window.

The scope of the variable in the parent package that is used in a parent variable configuration can be set to the Execute Package task, to the container that has the task, or to the package. If multiple variables with the same name are defined in a package, the variable that is closest in scope to the Execute Package task is used. The closest scope to the Execute Package task is the task itself.

To add a variable to a parent package

1. In Business Intelligence Development Studio, open the Integration Services project that contains the package to which you want to add a variable to pass to a child package.
2. In Solution Explorer, double-click the package to open it.
3. In SSIS Designer, to define the scope of the variable, do one of the following:
    - To set the scope to the package, click anywhere on the design surface of the **Control Flow** tab.
    - To set the scope to a parent container of the Execute Package task, click the container.
    - To set the scope to a parent container of the Execute Package task, click the task.
4. Add and configure a variable.

    Note:

    Select a data type that is compatible with the data that the variable will store.
5. To save the updated package, click **Save Selected Items** on the **File** menu.

To add a variable to a child package

1. In Business Intelligence Development Studio, open the Integration Services project that contains the package to which you want to add a parent variable configuration.
2. In Solution Explorer, double-click the package to open it.
3. In SSIS Designer, to set the scope to the package, click anywhere on the design surface of the **Control Flow** tab.
4. Add and configure a variable.

    Note:

    Select a data type that is compatible with the data that the variable will store.
5. To save the updated package, click **Save Selected Items** on the **File** menu.

To add a parent package configuration to a child package

1. If it is not already open, open the child package in Business Intelligence Development Studio.
2. Click anywhere on the design surface of the **Control Flow** tab.
3. On the **SSIS** menu, click **Package Configurations**.
4. In the **Package Configuration Organizer** dialog box, select **Enable package configuration**, and then click **Add**.
5. On the welcome page of the Package Configuration Wizard, click **Next.**
6. On the Select Configuration Type page, in the **Configuration type** list, select **Parent package variable** and do one of the following:
    - Select **Specify configuration settings directly**, and then in the **Parent variable** box, provide the name of the variable in the parent package to use in the configuration.

        Important:

        Variable names are case sensitive.

- Select or **Configuration location is stored in an environment variable,** and then in the **Environment variable list**, select the environment variable that contains the name of the variable.

7. Click **Next**.
8. On the Select Target Property page, expand the **Variable** node, and expand the **Properties** node of the variable to configure, and then click the property to be set by the configuration.
9. Click **Next**.
10. On the Completing the Wizard page, optionally, modify the default name of the configuration and review the configuration information.
11. Click **Finish** to complete the wizard and return to the **Package Configuration Organizer** dialog box.
12. In the **Package Configuration Organizer** dialog box, the **Configuration** box lists the new configuration.
13. Click **Close**.

http://technet.microsoft.com/en-us/library/ms345179(SQL.90).aspx

Q10 What is Execution Tree?

Execution Trees

Execution trees demonstrate how your package uses buffers and threads. At run time, the data flow engine breaks down Data Flow task operations into execution trees. These execution trees specify how buffers and threads are allocated in the package. Each tree creates a new buffer and may execute on a different thread. When a new buffer is created such as when a partially blocking or blocking transformation is added to the pipeline, additional memory is required to handle the data transformation; however, it is important to note that each new tree may also give you an additional worker thread.

Examine the execution trees in the example depicted in Figure 1 and Table 1 where two Employee datasets are combined together and then aggregated to load into a common destination table.

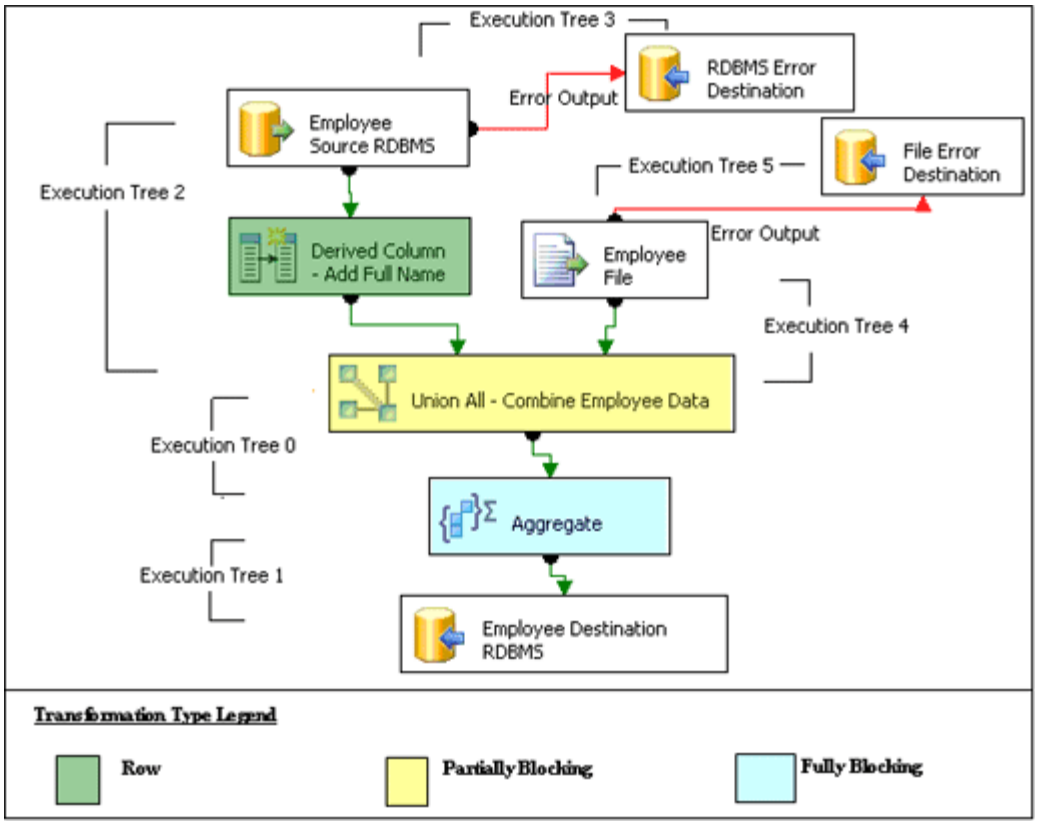


Figure 1: Example package

Note: Execution trees are listed in the table in the order that they execute.

Table 1: Execution trees defined

Execution Tree Enumeration	Description
begin **execution tree 2** output "OLE DB Source Output" (27) input "Derived Column Input" (172) output "Derived Column Output" (173) input "Union All Input 1" (411) output "Derived	In **Execution Tree 2**, SSIS reads data from the Employee OLE DB Source into the pipeline, a Derived Column transformation adds another column, and SSIS passes data to the Union All transformation. All of the operations in this execution tree use the same buffer; data is not copied again once it is read into the OLE DB Source Output.

Column Error Output" (174)

end execution tree 2

begin **execution tree 3**

output "OLE DB Source Error Output" (28)

input "OLE DB Destination Input" (2603)

In **Execution Tree 3**, SSIS creates a buffer to hold error records from the asynchronous Employee OLE DB Source before loading them into a destination error table.

output "OLE DB Destination Error Output" (2604)

end execution tree 3

begin **execution tree 4**

output "Flat File Source Output" (2363)

input "Union All Input 3" (2098)

In **Execution Tree 4**, SSIS reads data from the Employee Flat File Source and passes it to the Union All. These two operations use the same buffer.

end execution tree 4

begin **execution tree 5**

output "Flat File Source Error Output" (2364)

input "OLE DB Destination Input" (3818)

In **Execution Tree 5**, a buffer is created to hold errors from the asynchronous Employee Flat File Source before loading them into a destination error table.

output "OLE DB Destination Error Output" (3819)

end execution tree 5

begin **execution tree**

In **Execution Tree 0**, the Partially Blocking Union All transformation is

0

output "Union All Output 1" (412)

input "Aggregate Input 1" (2472)

executed and a new buffer is created to store the combined data and the aggregate is calculated.

end execution tree 0

begin **execution tree 1**

output "Aggregate Output 1" (2473)

input "OLE DB Destination Input" (150)

output "OLE DB Destination Error Output" (151)

In **Execution Tree 1**, after the Fully Blocking Aggregate transformation is completed, the output from the Aggregate operation is copied into a new buffer and data is loaded into the OLE DB Destination.

end execution tree 1

This example demonstrates how execution trees can help you understand buffer usage in a common SSIS package. This example also highlights how Partially Blocking transformations like Union All and Fully Blocking transformations like Aggregate create new buffers and threads whereas Row Transformations like Derived Column do not.

Execution trees are enormously valuable in understanding buffer usage. You can display execution trees for your own packages by turning on package logging, enabling logging for the Data Flow task, and then selecting the Pipeline Execution Tree event. Note that you will not see the execution trees until you execute the package. When you do execute the package, the execution trees appear in the Log Events window in Business Intelligence (BI) Development Studio.

http://technet.microsoft.com/en-us/library/cc966529.aspx

Q11 What are the points to keep in mind for performance improvement of the package?
http://technet.microsoft.com/en-us/library/cc966529.aspx

Q12 You may get a question stating a scenario and then asking you how would you create a package for that e.g. How would you configure a data flow task so that it can transfer data to different table based on the city name in a source table column?

Q13 Difference between Unionall and Merge Join?
I have been asked by many new SSIS developer about difference between Merge and Union all transformation in SSIS.

Well both of them essentially takes outputs from more than one sources and combines them into a single result set but there are couple of differences between two:

a) Merge transformation can accept only two inputs whereas Union all can take more than two inputs

b) Data has to be sorted before Merge Transformation whereas Union all doesn't have any condition like that.

http://sqlserversolutions.blogspot.com/2009/01/difference-between-merge-and-union-all.html

Q14 May get question regarding what X transformation do?Lookup, fuzzy lookup, fuzzy grouping transformation are my favorites.
For you.

Q15 How would you restart package from previous failure point?What are Checkpoints and how can we implement in SSIS?
Using Checkpoints in Packages

Integration Services can restart failed packages from the point of failure, instead of rerunning the whole package. If a package is configured to use checkpoints, information about 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 then re-created the next time the package is run.

Using checkpoints in a package can provide the following benefits.

- Avoid repeating the downloading and uploading of large files. For example, a package that downloads multiple large files by using an FTP task for each download can be restarted after the downloading of a single file fails and then download only that file.
- Avoid repeating the loading of large amounts of data. For example, a package that performs bulk inserts into dimension tables in a data warehouse using a different Bulk Insert task for each dimension can be restarted if the insertion fails for one dimension table, and only that dimension will be reloaded.
- Avoid repeating the aggregation of values. For example, a package that computes many aggregates, such as averages and sums, using a separate Data Flow task to perform each aggregation, can be restarted after computing an aggregation fails and only that aggregation will be recomputed.

If a package is configured to use checkpoints, Integration Services captures the restart point in the checkpoint file. The type of container that fails and the implementation of features such as transactions affect the restart point that is recorded in the checkpoint file. The current values of variables are also captured in the checkpoint file. However, the values of variables that have the **Object** data type are not saved in checkpoint files.

Defining Restart Points

The task host container, which encapsulates a single task, is the smallest atomic unit of work that can be restarted. The Foreach Loop container, the Data Flow task and all that it contains, and a transacted container are also treated as atomic units of work.

If a package is stopped while a transacted container is running, the transaction ends and any work performed by the container is rolled back. When the package is restarted, the container that failed is rerun. The completion of any child containers of the transacted container is not recorded in the checkpoint file. Therefore, when the package is restarted, the transacted container and its child containers run again.

Note:

Using checkpoints and transactions in the same package could cause unexpected results. For example, when a package fails and restarts from a checkpoint, the package might repeat a transaction that has already been successfully committed.

When a package is restarted from a checkpoint, the Foreach Loop container and its child containers are run again. If a child container in the loop previously ran successfully, this is not recorded in the checkpoint file; instead, the child container is run again.

If the package is restarted, the package configurations are not reloaded; instead the package uses the configuration information written to the checkpoint file. This ensures that, when the package is run again, the package uses the same configurations as when it failed.

A package can be restarted only at the control flow level. You cannot restart a package in the middle of a data flow. To avoid rerunning the whole data flow, the package might be designed to include multiple Data Flow tasks. This way the package can be restarted, and will rerun only the Data Flow tasks that failed.

Configuring a Package to Restart

The checkpoint file includes the execution results of all completed containers, the current values of system and user-defined variables, and package configuration information. The file also includes the unique identifier of the package. To successfully restart a package, the package identifier in the checkpoint file and the package must match; otherwise the restart fails. This prevents a package from using a checkpoint file written by a different package version. If the package runs successfully, after it is restarted the checkpoint file is deleted.

The following table lists the package properties that you set to implement checkpoints.

Property	Description
CheckpointFileName	Specifies the name of the checkpoint file.
CheckpointUsage	Specifies whether checkpoints are used.
SaveCheckpoints	Indicates whether the package saves checkpoints. This property must be set to True to restart a package from a point of failure.

Additionally, you must set the **FailPackageOnFailure** property to **true** for all the containers in the package that you want to identify as restart points.

You can use the **ForceExecutionResult** property to test the use of checkpoints in a package. By setting **ForceExecutionResult** of a task or container to Failure, you can imitate real-time failure. When you rerun the package, the failed task and containers will be rerun.

Checkpoint Usage

The **CheckpointUsage** property can be set to the following values:

Value	Description
Never	Specifies that the checkpoint file is not used and that the package runs from the start of the package workflow.
Always	Specifies that the checkpoint file is always used and that the package restarts from the point of the previous execution failure. If the checkpoint file is not found, the package fails.
IfExists	Specifies that the checkpoint file is used if it exists. 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.

Note:

The **/CheckPointing on** option of dtexec is equivalent to setting the **SaveCheckpoints** property of the package to **True**, and the **CheckpointUsage** property to Always. For more information, see dtexec Utility.

Securing Checkpoint Files

Package level protection does not include protection of checkpoint files and you must secure these files separately. Checkpoint data can be stored only in the file system and you should use an operating system access control list (ACL) to secure the location or folder where you store the file. It is important to secure checkpoint files because they contain information about the package state, including the current values of variables. For example, a variable may contain a recordset with many rows of private data such as telephone numbers. For more information, see Controlling Access to Files Used by Packages.

To configure the checkpoint properties

- How to: Configure Checkpoints for Restarting a Failed Package

http://msdn.microsoft.com/en-us/library/ms140226.aspx

Q16 Where are SSIS package stored in the SQL Server?
MSDB.sysdtspackages90 stores the actual content and ssydtscategories, sysdtslog90, sysdtspackagefolders90, sysdtspackagelog, sysdtssteplog, and sysdtstasklog do the supporting roles.

Q17 How would you schedule a SSIS packages?
Using SQL Server Agent. Read about Scheduling a job on Sql server Agent

Q18 Difference between asynchronous and synchronous transformations?
Asynchronous transformation have different Input and Output buffers and it is up to the component designer in an Async component to provide a column structure to the output buffer and hook up the data from the input.

Q19 How to achieve multiple threading in SSIS?

Source:
http://sqlserversolutions.blogspot.com/2009/02/ssis-interview-questions.html

---

1) What is the control flow
2) what is a data flow
3) how do you do error handling in SSIS
4) how do you do logging in ssis
5) how do you deploy ssis packages.
6) how do you schedule ssis packages to run on the fly
7) how do you run stored procedure and get data
8) A scenario: Want to insert a tect file into database table, but during the upload want to change a column called as months - January, Feb, etc to a code, - 1,2,3.. .This code can be read from another database table called months. After the conversion of the data , upload the file. If there are any errors, write to error table. Then for all errors, read errors from database, create a file, and mail it to the supervisor.
How would you accomplish this task in SSIS?
9)what are variables and what is variable scope ?

The website also says 'These are SSIS fundamentals and if you want to be a competent developer those are the MINIMUM that you need to know...'

For Q 1 and 2:
In SSIS a workflow is called a control-flow. A control-flow links together our modular data-flows as a series of operations in order to achieve a desired result.

A control flow consists of one or more tasks and containers that execute when the package runs. To control order or define the conditions for running the next task or container in the package control flow, you use precedence constraints to connect the tasks and containers in a package. A subset of tasks and containers can also be grouped and run repeatedly as a unit within the package control flow.

SQL Server 2005 Integration Services (SSIS) provides three different types of control flow elements: containers that provide structures in packages, tasks that provide functionality, and precedence constraints that connect the executables, containers, and tasks into an ordered control flow.

A data flow consists of the sources and destinations that extract and load data, the transformations that modify and extend data, and the paths that link sources, transformations, and destinations. Before you can add a data flow to a package, the package control flow must include a Data Flow task. The Data Flow task is the executable within the SSIS package that creates, orders, and runs the data flow. A separate instance of the data flow engine is opened for each Data Flow task in a package.

SQL Server 2005 Integration Services (SSIS) provides three different types of data flow components: sources, transformations, and destinations. Sources extract data from data stores

such as tables and views in relational databases, files, and Analysis Services databases. Transformations modify, summarize, and clean data. Destinations load data into data stores or create in-memory datasets.

Q3:

When a data flow component applies a transformation to column data, extracts data from sources, or loads data into destinations, errors can occur. Errors frequently occur because of unexpected data values.

For example, a data conversion fails because a column contains a string instead of a number, an insertion into a database column fails because the data is a date and the column has a numeric data type, or an expression fails to evaluate because a column value is zero, resulting in a mathematical operation that is not valid.

Errors typically fall into one the following categories:

-Data conversion errors, which occur if a conversion results in loss of significant digits, the loss of insignificant digits, and the truncation of strings. Data conversion errors also occur if the requested conversion is not supported.
-Expression evaluation errors, which occur if expressions that are evaluated at run time perform invalid operations or become syntactically incorrect because of missing or incorrect data values.
-Lookup errors, which occur if a lookup operation fails to locate a match in the lookup table.

Many data flow components support error outputs, which let you control how the component handles row-level errors in both incoming and outgoing data. You specify how the component behaves when truncation or an error occurs by setting options on individual columns in the input or output.

For example, you can specify that the component should fail if customer name data is truncated, but ignore errors on another column that contains less important data.

Q 4:

SSIS includes logging features that write log entries when run-time events occur and can also write custom messages.

Integration Services supports a diverse set of log providers, and gives you the ability to create custom log providers. The Integration Services log providers can write log entries to text files, SQL Server Profiler, SQL Server, Windows Event Log, or XML files.

Logs are associated with packages and are configured at the package level. Each task or container in a package can log information to any package log. The tasks and containers in a package can be enabled for logging even if the package itself is not.

To customize the logging of an event or custom message, Integration Services provides a schema of commonly logged information to include in log entries. The Integration Services log schema defines the information that you can log. You can select elements from the log schema for each log entry.

To enable logging in a package
1. In Business Intelligence Development Studio, open the Integration Services project that contains the package you want.
2. On the SSIS menu, click Logging.
3. Select a log provider in the Provider type list, and then click Add.

Q 5 :

SQL Server 2005 Integration Services (SSIS) makes it simple to deploy packages to any computer. There are two steps in the package deployment process:
-The first step is to build the Integration Services project to create a package deployment utility.
-The second step is to copy the deployment folder that was created when you built the Integration Services project to the target computer, and then run the Package Installation Wizard to install the packages.

Q 9 :

Variables store values that a SSIS 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. The precedence constraints that sequence tasks and containers into a workflow can use variables when their constraint definitions include expressions.

Integration Services supports two types of variables: user-defined variables and system variables. User-defined variables are defined by package developers, and system variables are defined by Integration Services. You can create as many user-defined variables as a package requires, but you cannot create additional system variables.

Scope :

A variable is created within the scope of a package or within the scope of a container, task, or event handler in the package. Because the package container is at the top of the container hierarchy, variables with package scope function like global variables and can be used by all containers in the package. Similarly, variables defined within the scope of a container such as a For Loop container can be used by all tasks or containers within the For Loop container.

More to come...

Here are some more SSIS related Interview Questions which I got from dotnetspider.
Hope they help.

Question 1 - True or False - Using a checkpoint file in SSIS is just like issuing the CHECKPOINT command against the relational engine. It commits all of the data to the database.
False. SSIS provides a Checkpoint capability which allows a package to restart at the point of failure.

Question 2 - Can you explain the what the Import\Export tool does and the basic steps in the wizard?
The Import\Export tool is accessible via BIDS or executing the dtswizard command.
The tool identifies a data source and a destination to move data either within 1 database, between instances or even from a database to a file (or vice versa).

Question 3 - What are the command line tools to execute SQL Server Integration Services packages?
DTSEXECUI - When this command line tool is run a user interface is loaded in order to configure each of the applicable parameters to execute an SSIS package.
DTEXEC - This is a pure command line tool where all of the needed switches must be passed into the command for successful execution of the SSIS package.

Question 4 - Can you explain the SQL Server Integration Services functionality in Management Studio?
You have the ability to do the following:
Login to the SQL Server Integration Services instance
View the SSIS log
View the packages that are currently running on that instance
Browse the packages stored in MSDB or the file system
Import or export packages
Delete packages
Run packages

Question 5 - Can you name some of the core SSIS components in the Business Intelligence Development Studio you work with on a regular basis when building an SSIS package?
Connection Managers
Control Flow
Data Flow
Event Handlers
Variables window
Toolbox window
Output window
Logging
Package Configurations

Question Difficulty = Moderate

Question 1 - True or False: SSIS has a default means to log all records updated, deleted or inserted on a per table basis.
False, but a custom solution can be built to meet these needs.

Question 2 - What is a breakpoint in SSIS? How is it setup? How do you disable it?
A breakpoint is a stopping point in the code. The breakpoint can give the Developer\DBA an opportunity to review the status of the data, variables and the overall status of the SSIS package. 10 unique conditions exist for each breakpoint.
Breakpoints are setup in BIDS. In BIDS, navigate to the control flow interface. Right click on the object where you want to set the breakpoint and select the 'Edit Breakpoints...' option.

Question 3 - Can you name 5 or more of the native SSIS connection managers?
OLEDB connection - Used to connect to any data source requiring an OLEDB connection (i.e., SQL Server 2000)
Flat file connection - Used to make a connection to a single file in the File System. Required for reading information from a File System flat file
ADO.Net connection - Uses the .Net Provider to make a connection to SQL Server 2005 or other connection exposed through managed code (like C#) in a custom task
Analysis Services connection - Used to make a connection to an Analysis Services database or project. Required for the Analysis Services DDL Task and Analysis Services Processing Task
File connection - Used to reference a file or folder. The options are to either use or create a file or folder
Excel

FTP
HTTP
MSMQ
SMO
SMTP
SQLMobile
WMI

Question 4 - How do you eliminate quotes from being uploaded from a flat file to SQL Server?
In the SSIS package on the Flat File Connection Manager Editor, enter quotes into the Text qualifier field then preview the data to ensure the quotes are not included.
Additional information: How to strip out double quotes from an import file in SQL Server Integration Services
Question 5 - Can you name 5 or more of the main SSIS tool box widgets and their functionality?
For Loop Container
Foreach Loop Container
Sequence Container
ActiveX Script Task
Analysis Services Execute DDL Task
Analysis Services Processing Task
Bulk Insert Task
Data Flow Task
Data Mining Query Task
Execute DTS 2000 Package Task
Execute Package Task
Execute Process Task
Execute SQL Task
etc.

Question Difficulty = Difficult

Question 1 - Can you explain one approach to deploy an SSIS package?
One option is to build a deployment manifest file in BIDS, then copy the directory to the applicable SQL Server then work through the steps of the package installation wizard
A second option is using the dtutil utility to copy, paste, rename, delete an SSIS Package
A third option is to login to SQL Server Integration Services via SQL Server Management Studio then navigate to the 'Stored Packages' folder then right click on the one of the children folders or an SSIS package to access the 'Import Packages...' or 'Export Packages...'option.
A fourth option in BIDS is to navigate to File | Save Copy of Package and complete the interface.

Question 2 - Can you explain how to setup a checkpoint file in SSIS?
The following items need to be configured on the properties tab for SSIS package:
CheckpointFileName - Specify the full path to the Checkpoint file that the package uses to save the value of package variables and log completed tasks. Rather than using a hard-coded path as shown above, it's a good idea to use an expression that concatenates a path defined in a package variable and the package name.
CheckpointUsage - Determines if/how checkpoints are used. Choose from these options: Never (default), IfExists, or Always. Never indicates that you are not using Checkpoints. IfExists is the

typical setting and implements the restart at the point of failure behavior. If a Checkpoint file is found it is used to restore package variable values and restart at the point of failure. If a Checkpoint file is not found the package starts execution with the first task. The Always choice raises an error if the Checkpoint file does not exist.
SaveCheckpoints - Choose from these options: True or False (default). You must select True to implement the Checkpoint behavior.

Question 3 - Can you explain different options for dynamic configurations in SSIS?
Use an XML file
Use custom variables
Use a database per environment with the variables
Use a centralized database with all variables

Question 4 - How do you upgrade an SSIS Package?
Depending on the complexity of the package, one or two techniques are typically used:
Recode the package based on the functionality in SQL Server DTS
Use the Migrate DTS 2000 Package wizard in BIDS then recode any portion of the package that is not accurate

Question 5 - Can you name five of the Perfmon counters for SSIS and the value they provide?
SQLServer:SSIS Service
SSIS Package Instances - Total number of simultaneous SSIS Packages running
SQLServer:SSIS Pipeline
BLOB bytes read - Total bytes read from binary large objects during the monitoring period.
BLOB bytes written - Total bytes written to binary large objects during the monitoring period.
BLOB files in use - Number of binary large objects files used during the data flow task during the monitoring period.
Buffer memory - The amount of physical or virtual memory used by the data flow task during the monitoring period.
Buffers in use - The number of buffers in use during the data flow task during the monitoring period.
Buffers spooled - The number of buffers written to disk during the data flow task during the monitoring period.
Flat buffer memory - The total number of blocks of memory in use by the data flow task during the monitoring period.
Flat buffers in use - The number of blocks of memory in use by the data flow task at a point in time.
Private buffer memory - The total amount of physical or virtual memory used by data transformation tasks in the data flow engine during the monitoring period.
Private buffers in use - The number of blocks of memory in use by the transformations in the data flow task at a point in time.
Rows read - Total number of input rows in use by the data flow task at a point in time.
Rows written - Total number of output rows in use by the data flow task at a point in time.

Source:

http://forums.keysoft.co.in/forum_posts.asp?TID=47

SQL Server Integration Services (SSIS) Interview questions 

1. What is for-loop container? Give an example of where it can be used.
2. What is foreach-loop container? Give an example of where it can be used.
3. What is sequence container? Give an example of where it can be used.
4. What is the difference between Analysis Services processing task & Analysis services execute DDL task?
5. What is the difference between for-loop container & foreach-loop container?
6. What are the different parameters or configurations that "send mail task" requires?
7. Mention few mapping operations that the Character Map transformation supports.
8. Explain the functionality of: Import Column Transformation and Export Column Transformation
9. Explain the functionality of: Percentage Sampling transformation
10. Explain the functionality of: SCD transformation
11. Explain the functionality of: Union All transformation
12. What does "Lookup" transformation used for?
13. What are checkpoints? For which objects we define checkpoint? How to configure checkpoint for a package?
14. What is the use of "package configurations" available in SSIS?
15. What are the different ways in which configuration details can be stored?
16. How to deploy a package from development server to production server?
17. How to create Integration Services Package Deployment Utility?
18. How to deploy packages to file system?
19. How to deploy packages to SQL server? Where in database packages will be stored?
20. How to set security for a package? Explain the same as per different deployment options.
21. Explain the architecture of SSIS
22. Explain the how SSIS engine workflow

Source:

http://www.datawarehousingguide.com/content/view/95/60/

Microsoft Business Intelligence frequently asked questions 

Q. What is Business Intelligence and what does it do?

Business Intelligence, a complete suite of server, client, and developer applications fully integrated with the 2007 Microsoft Office system, delivers business intelligence on the desktop in an integrated, centrally managed environment.

Business Intelligence simplifies information discovery and analysis, making it possible for decision-makers at all levels of an organization to more easily access, understand, analyze, collaborate, and act on information, anytime and anywhere. Move from just consuming information to developing deep contextual knowledge about that information. By tying strategy

to metrics, organizations can gain competitive advantage by making better decisions faster, at all levels of the organization.

Business Intelligence delivers business intelligence to everyone in an organization by integrating two major components:

* **The Business Intelligence platform**, driven by Microsoft SQL Server 2005 and including its powerful relational database management system, SQL Server Integration Services, SQL Server Analysis Services, SQL Server Reporting Services, and SQL Server Data Mining capabilities. Business Intelligence is built on the scalable and reliable SQL Server 2005 platform, proven to support mission-critical environments, and integrated with the Microsoft Visual Studio 2005 development platform.

The 2007 Microsoft Office system, delivering information through the tools that users already know and rely on. Users can share more powerful, interactive spreadsheets using improved charting and formula authoring, greater row and column capacity, and enhanced sorting and filtering along with enhanced PivotTable and PivotChart views. With server-based spreadsheets, you can share information broadly with confidence, knowing that your information is more secure and centrally managed, yet accessible to colleagues, customers, and partners through the Web. Dynamic scorecards combine the power of predictive analysis with real-time reporting. Strategy maps make it easy to visualize key areas — you can see trends, identify problem areas early, maximize success areas, and monitor performance against key goals in real time.

Q. Who is Business Intelligence for?

Business Intelligence is for businesses that want to drive intelligent decision-making throughout their organizations and make it easy for everyone in the organization to collaborate, analyze, share, and act on business information from a centrally managed, more secure source. Enterprise grade yet attractively priced, Business Intelligence supports IT professionals, information workers, and developers, and empowers organizations of all sizes.

Q. What if I have a small (fewer than 100-person) company? Can I still use Business Intelligence?

Yes. Business Intelligence provides an excellent business intelligence solution for organizations of all sizes. You can deploy reporting solutions to a small workgroup or department with SQL Server 2005 Reporting Services. You can also perform queries and analysis using Excel Services — new to the 2007 release of Microsoft Office — through Microsoft Office SharePoint Server 2007. This combination delivers Web-based query and analysis capabilities to every user in a format that is easy to use and centrally secured and managed.

Q. What's a typical way an organization might use Business Intelligence?

Business Intelligence connects the right people to the right information at the right time. For example, when reviewing the current financial scorecard, your sales manager, Margaret, notices that one particular region is not contributing as much as other regions. When analyzing the data from the spreadsheet for the low-performing region, she notices that one particular salesperson,

Joe, has below-average sales numbers.

At the same time, Joe receives through e-mail a weekly status report that contains qualified leads in the region, pipeline information, and details about deals closed. Next, he opens the dashboard and searches on information about his top account, and he sees data from his enterprise resource planning (ERP) system related to that account. Joe notes that his average deal size is smaller than others in the region. It's easy for Joe to find out why this is by doing some "what if" analysis. He inputs different variables to determine the number of leads he needs to reach the company sales average. Next, by doing further analysis on data for the region, Joe can compare his sales numbers with regional averages. He adds more information that shows the discount rate, and then adds visualization to better understand the results. The visual representation of the data shows Joe that his discount rate is much lower than the average for the region.

Next, it's time to tell his manager. Joe publishes this information to the server, schedules a meeting with Margaret to discuss getting approval to increase the discount rate so that he'll be better able to compete, and alerts Margaret through online collaboration that he's just posted his analysis report.

Joe and Margaret meet to discuss details. Afterward, he makes a note on the key performance indicator (KPI) that he owns for that region, and Margaret sees the annotation in her latest scorecard as a reminder that there's a new strategy in place to increase Joe's results and address the poor sales performance.

What programs are included in Business Intelligence? Is Business Intelligence available as a single product in a box?

Business Intelligence includes two major components: the business intelligence platform (SQL Server 2005) and end-user tools — the 2007 Microsoft Office system.

Source:

http://www.datawarehousingguide.com/content/view/60/60/

What is ETL(Extract, Transform, and Load) ?

ETL stands for extract, transform and load, the processes that enable companies to move data from multiple sources, reformat and cleanse it, and load it into another database, a data mart or a data warehouse for analysis, or on another operational system to support a business process.

ETL - Table of contents

- What is ETL?
- Extraction
- Transformation

- Loading
- Challenges of ETL
- Tools in market

What is ETL (Extract, Transform, and Load) ?

Extract, Transform, and Load (ETL) is a process in data warehousing that involves

- extracting data from outside sources,
- transforming it to fit business needs (which can include quality levels), and ultimately
- loading it into the end target, i.e. the data warehouse.

ETL is important, as it is the way data actually gets loaded into the warehouse. This article assumes that data is always loaded into a data warehouse, whereas the term ETL can in fact refer to a process that loads any database. ETL can also be used for the integration with legacy systems. Usually ETL implementations store an audit trail on positive and negative process runs. In almost all designs, this audit trail is not at the level of granularity which would allow to reproduce the ETL's result if the raw data were not available.

Extraction

The first part of an ETL process is to extract the data from the source systems. Most data warehousing projects consolidate data from different source systems. Each separate system may also use a different data organization / format. Common data source formats are relational databases and flat files, but may include non-relational database structures such as IMS or other data structures such as VSAM or ISAM, or even fetching from outside sources such as web spidering or screen-scraping. Extraction converts the data into a format for transformation processing.

An intrinsic part of the extraction is the parsing of extracted data, resulting in a check if the data meets an expected pattern or structure. If not, the data is rejected entirely.

Transformation

The transform stage applies a series of rules or functions to the extracted data from the source to derive the data to be loaded to the end target. Some data sources will require very little or even no manipulation of data. In other cases, one or more of the following transformations types to meet the business and technical needs of the end target may be required:

- Selecting only certain columns to load (or selecting null columns not to load)
- Translating coded values (e.g., if the source system stores 1 for male and 2 for female, but the warehouse stores M for male and F for female), this is called automated data cleansing; no manual cleansing occurs during ETL
- Encoding free-form values (e.g., mapping "Male" to "1" and "Mr" to M)
- Deriving a new calculated value (e.g., sale_amount = qty * unit_price)
- Joining together data from multiple sources (e.g., lookup, merge, etc.)
- Summarizing multiple rows of data (e.g., total sales for each store, and for each region)
- Generating surrogate key values

- Transposing or pivoting (turning multiple columns into multiple rows or vice versa)
- Splitting a column into multiple columns (e.g., putting a comma-separated list specified as a string in one column as individual values in different columns)
- Applying any form of simple or complex data validation; if failed, a full, partial or no rejection of the data, and thus no, partial or all the data is handed over to the next step, depending on the rule design and exception handling. Most of the above transformations itself might result in an exception, e.g. when a code-translation parses an unknown code in the extracted data.

Loading

The load phase loads the data into the end target, usually being the data warehouse (DW). Depending on the requirements of the organization, this process ranges widely. Some data warehouses might weekly overwrite existing information with cumulative, updated data, while other DW (or even other parts of the same DW) might add new data in a historized form, e.g. hourly. The timing and scope to replace or append are strategic design choices dependent on the time available and the business needs. More complex systems can maintain a history and audit trail of all changes to the data loaded in the DW.

As the load phase interacts with a database, the constraints defined in the database schema as well as in triggers activated upon data load apply (e.g. uniqueness, referential integrity, mandatory fields), which also contribute to the overall data quality performance of the ETL process

Challenges and Complexities in ETL process

ETL processes can be quite complex, and significant operational problems can occur with improperly designed ETL systems.

The range of data values or data quality in an operational system may be outside the expectations of designers at the time validation and transformation rules are specified. Data profiling of a source during data analysis is recommended to identify the data conditions that will need to be managed by transform rules specifications. This will lead to an amendment of validation rules explicitly and implicitly implemented in the ETL process.

DW are typically fed asynchronously by a variety of sources which all serve a different purpose, resulting in e.g. different reference data. ETL is a key process to bring heterogeneous and asynchronous source extracts to a homogeneous environment.

The scalability of an ETL system across the lifetime of its usage, needs to be established during analysis. This includes understanding the volumes of data that will have to be processed within service level agreements (SLAs). The time available to extract from source systems may change, which may mean the same amount of data may have to be processed in less time. Some ETL systems have to scale to process terabytes of data to update data warehouses with tens of terabytes of data. Increasing volumes of data may require designs that can scale from daily batch to intra-day micro-batch to integration with message queues or real-time change data capture (CDC) for continuous transformation and update.

ETL tools in the market

While an ETL process can be created using almost any programming language, creating them from scratch is quite complex. Increasingly, companies are buying ETL tools to help in the

creation of ETL processes.

By using an established ETL framework, you are more likely to end up with better connectivity and scalability. A good ETL tool must be able to communicate with the many different relational databases and read the various file formats used throughout an organization. ETL tools have started to migrate into Enterprise Application Integration, or even Enterprise Service Bus, systems that now cover much more than just the extraction, transformation and loading of data. Many ETL vendors now have data profiling, data quality and metadata capabilities.

Some of the well known ETL tools are Informatica, Ab initio, SSIS, datastage, Pentaho kettle and more.

Source:
http://www.datawarehousingguide.com/content/view/116/66/

Thanks
Ramu Ragavan
Thomson Reuters
Email: ragavan.ramu@thomsonreuters.com
M- 9343992012