

**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?**

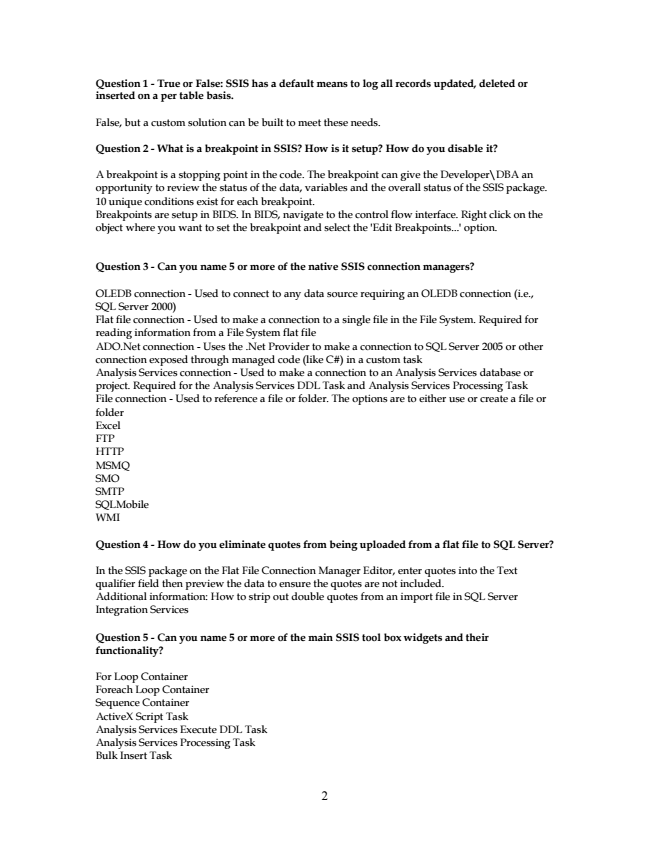
You have the ability to do the following: Login to the SQL Server Integration Services instance View the SSIS log View the packages that are currently running on that instance Browse the packages stored in MSDB or the file system Import or export packages Delete packages Run packages

**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**

1



**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

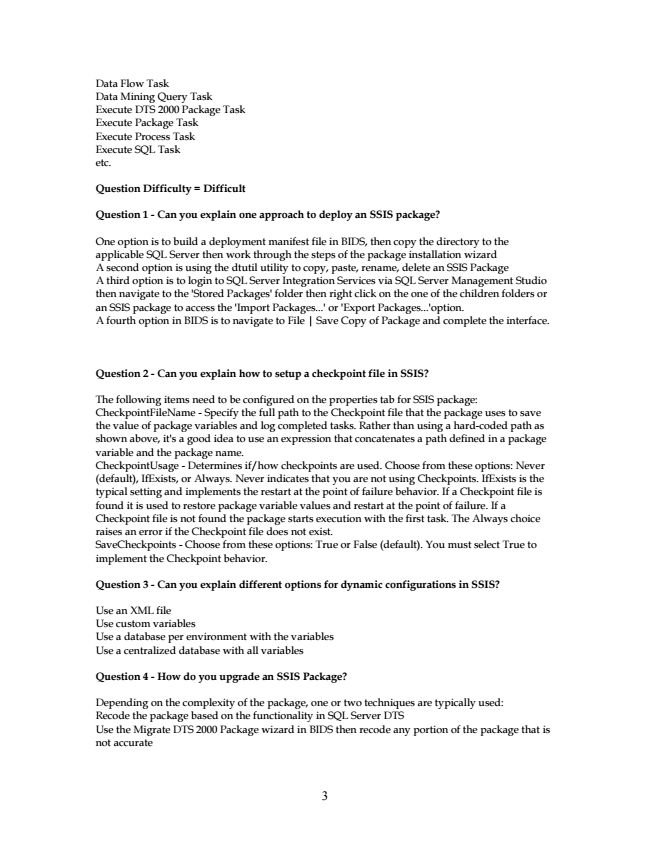
**Question 4 - How do you eliminate quotes from being uploaded from a flat file to SQL Server?**

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

2



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.

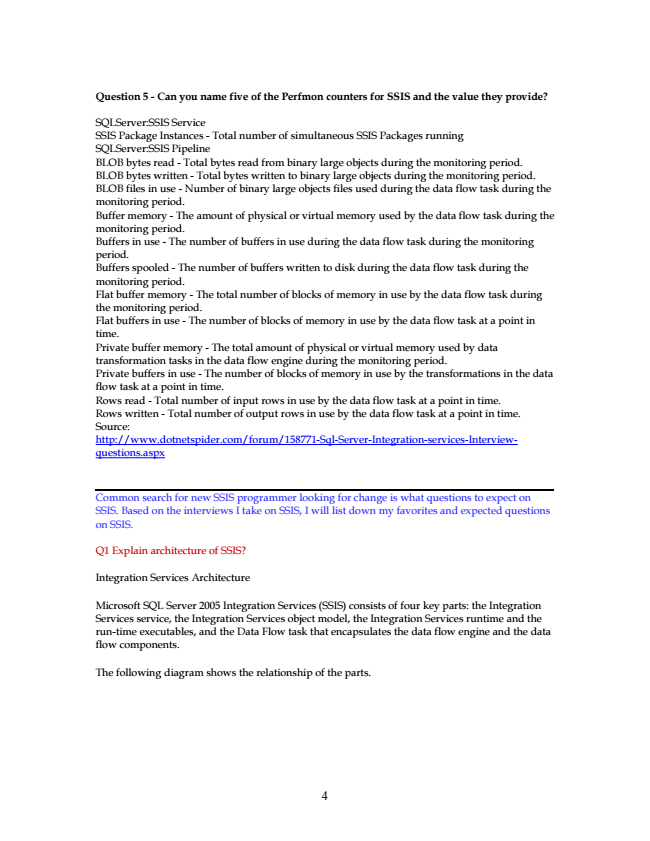
**Question 3 - Can you explain different options for dynamic configurations in SSIS?**

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

3



**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.

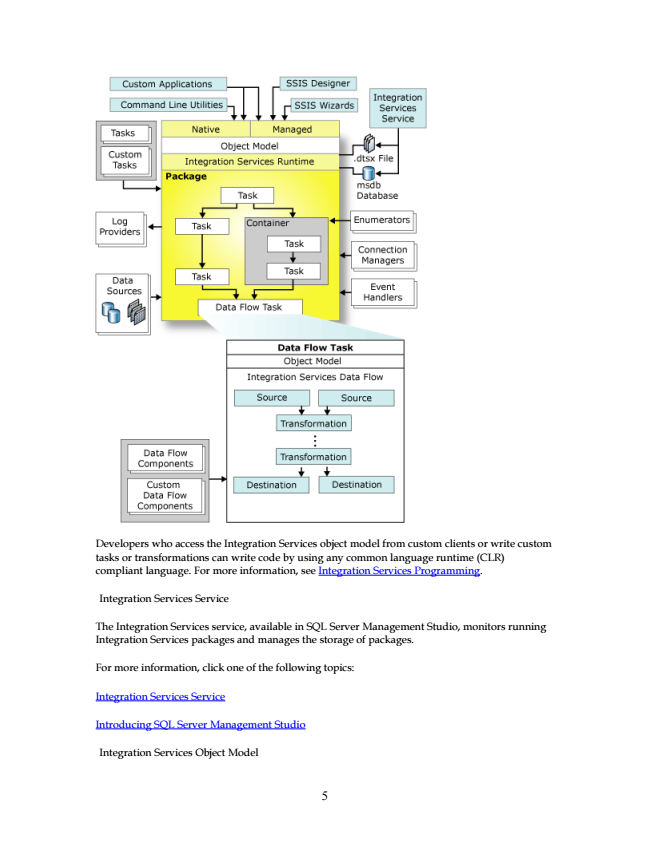
Q1 Explain architecture of SSIS?

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.

4



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.

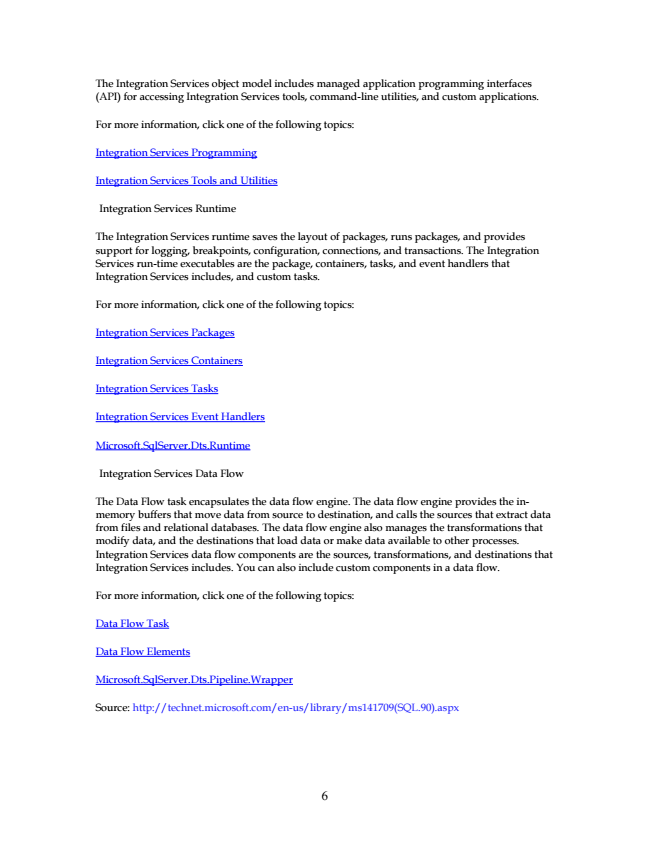
For more information, click one of the following topics:

Integration Services Service

Introducing SQL Server Management Studio

Integration Services Object Model

5



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:

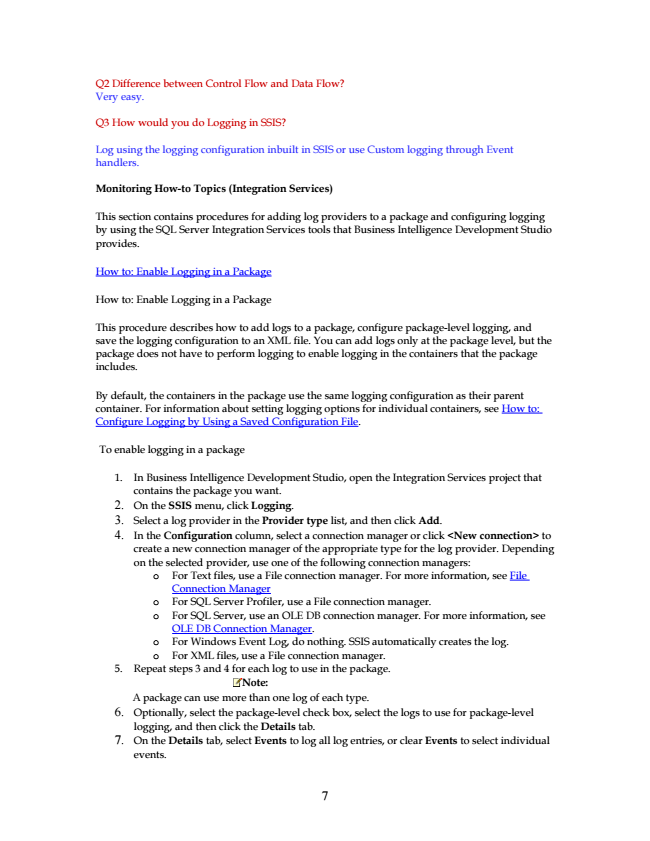
Data Flow Task

Data Flow Elements

Microsoft.SqlServer.Dts.Pipeline.Wrapper

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

6



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:

o For Text files, use a File connection manager. For more information, see File

Connection Manager o For SQL Server Profiler, use a File connection manager. o For SQL Server, use an OLE DB connection manager. For more information, see

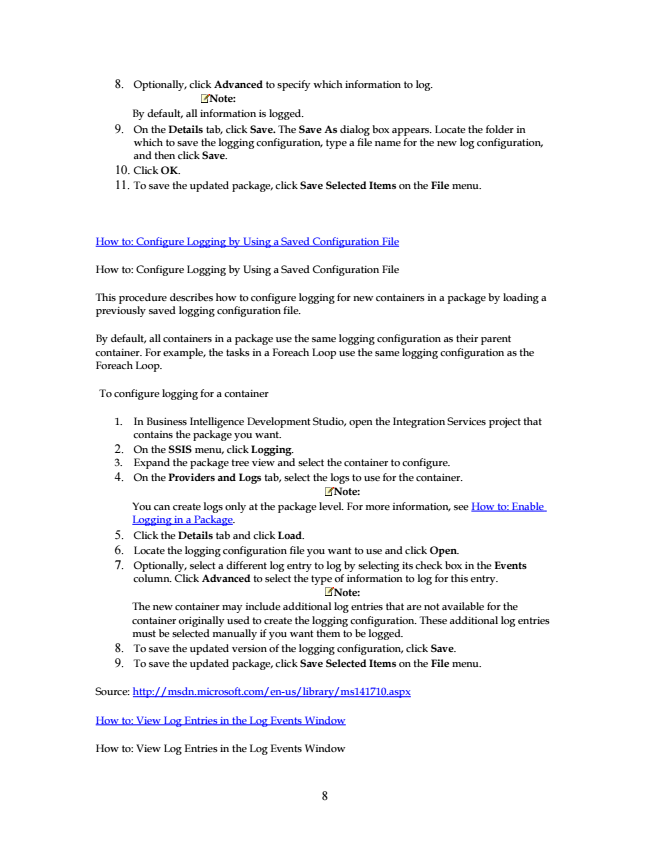
OLE DB Connection Manager. o For Windows Event Log, do nothing. SSIS automatically creates the log. o For XML files, use a File connection manager. 5. Repeat steps 3 and 4 for each log to use in the package.

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.

7



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.

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

8

To configure logging for a container

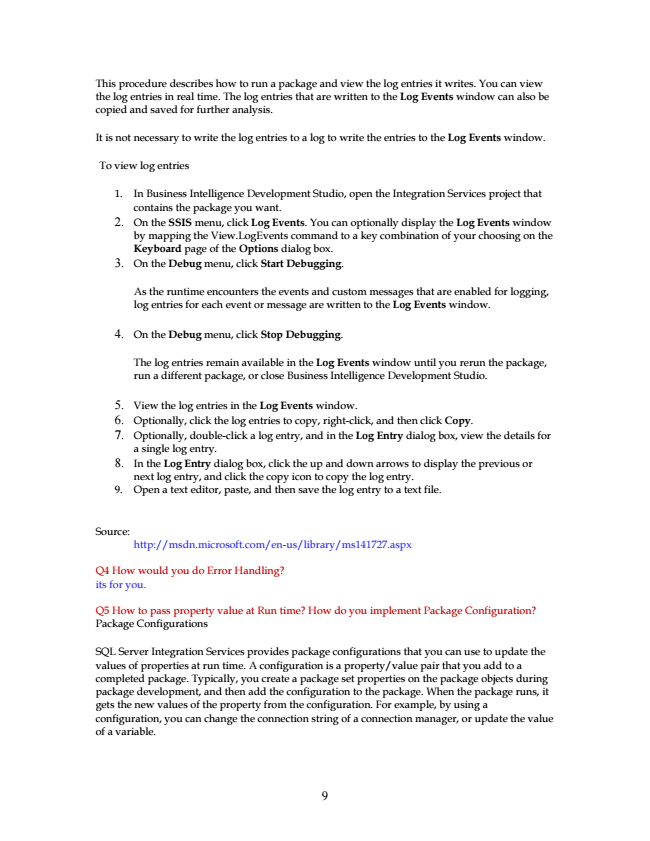
1. In Business Intelligence Development Studio, open the Integration Services project that

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.

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

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.



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:

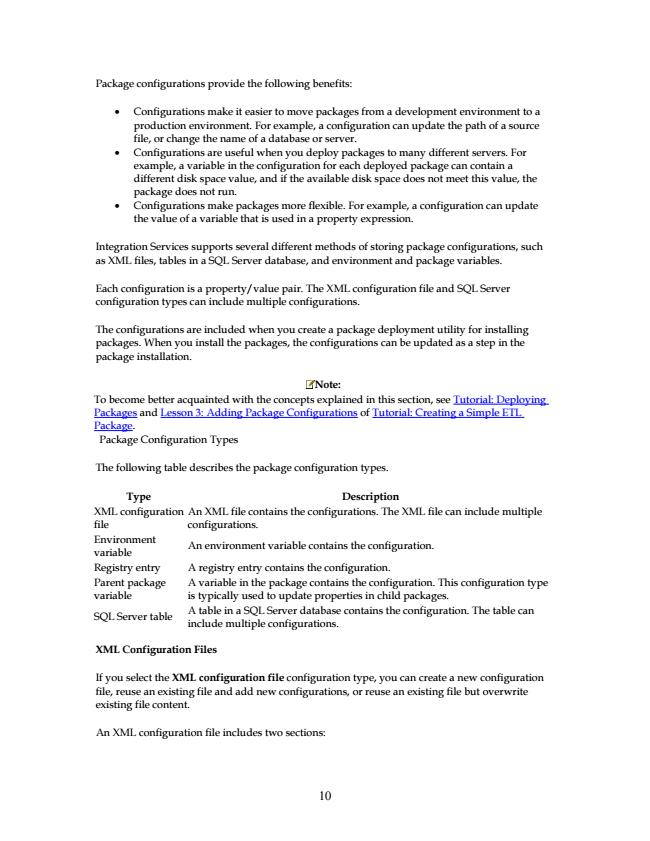
http://msdn.microsoft.com/en-us/library/ms141727.aspx

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.

9



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:

10

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.