SSIS,SSAS,SSRS and BI ROAD AHEAD

SSIS

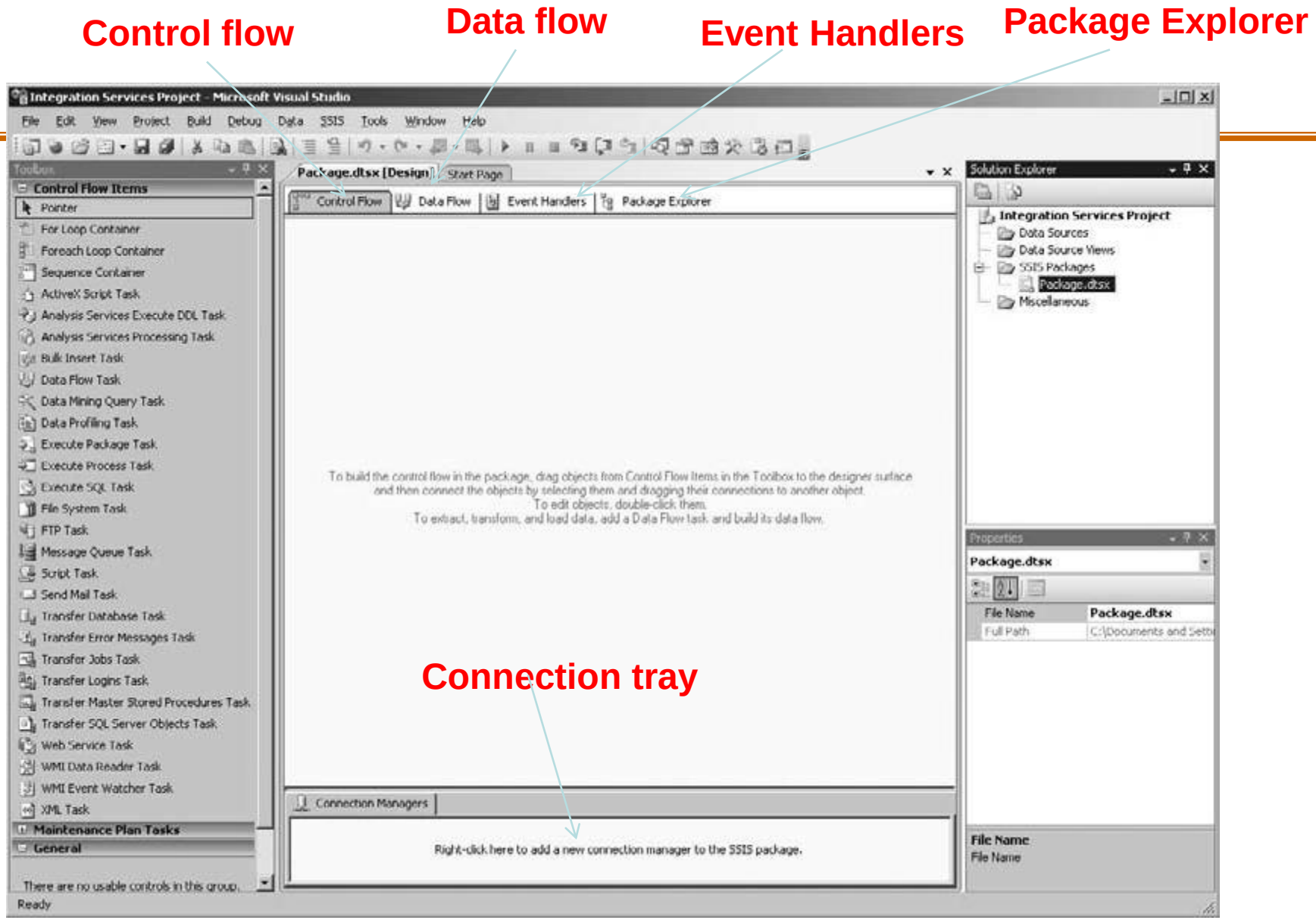
- Integration Services create structures called packages, which are used to move data between systems.
- A package's overall operation is defined by the control flow. The *control flow* is the sequence of tasks that will be performed by the package. These may be tasks such as transferring a file using File Transfer Protocol (FTP), logging an event, truncating a table in preparation for a data load, or even e-mailing a status message to an administrator. Of course, one of the most-used control flow tasks is the *data flow task,* which moves data from one place to another.

SSIS

- The data flow task provides the plumbing for moving data from one or more sources to one or more destinations. Along the way, the data may be subjected to any number of transformation processes. Data can now be manipulated as it is flowing through “the pipeline” without having to be deposited in a temporary, working table.

SSIS

- Integration Services packages are created using an Integration Services project in the Business Intelligence Development Studio. When we are working on an Integration Services project, the Designer window contains four tabs, as shown in Figure.


Control flow
Data flow
Event Handlers
Package Explorer
Connection tray
Integration Services Project - Microsoft Visual Studio
File Edit View Project Build Debug Data SSIS Tools Window Help
Control Flow Items
Pointer
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
Data Profiling Task
Execute Package Task
Execute Process Task
Execute SQL Task
File System Task
FTP Task
Message Queue Task
Script Task
Send Mail Task
Transfer Database Task
Transfer Error Messages Task
Transfer Jobs Task
Transfer Logins Task
Transfer Master Stored Procedures Task
Transfer SQL Server Objects Task
Web Service Task
WMI Data Reader Task
WMI Event Watcher Task
XML Task
Maintenance Plan Tasks
General
There are no usable controls in this group.
Package.dtsx [Design]
Start Page
Control Flow
Data Flow
Event Handlers
Package Explorer
To build the control flow in the package, drag objects from Control Flow items in the Toolbox to the designer surface and then connect the objects by selecting them and dragging their connections to another object.
To edit objects, double-click them.
To extract, transform, and load data, add a Data Flow task, and build its data flow.
Connection Managers
Right-click here to add a new connection manager to the SSIS package.
Solution Explorer
Integration Services Project
Data Sources
Data Source Views
SSIS Packages
Package.dtsx
Miscellaneous
Properties
Package.dtsx
File Name
Package.dtsx
Full Path
File Name
File Name
Ready

SSIS

- The first three tabs—the Control Flow Designer tab, the Data Flow Designer tab, and the Event Handlers Designer tab—let us define various types of functionality for the package.
- The final tab—the Package Explorer—provides an alternative view of the contents of the package. In addition to the tabs, there is a special area at the bottom of the Designer window for defining Connection Managers, which is called the *Connections tray.*

SSIS

- Each Integration Services package contains a control flow to define the overall operation of the package. The control flow is defined by dragging items from the Toolbox onto the Control Flow Designer tab. When the Control Flow Designer tab is selected, only those tasks that can be used on this tab are displayed in the Toolbox

SSIS

- An Integration Services package may contain several data flows. Each data flow is represented by a data flow task placed on the Control Flow Designer tab

SSIS

- Integration Services packages are event-driven. This means we can specify routines to execute when a particular event occurs. An *event* can be the completion of a task or an error that occurs during task execution. The routine for the event is known as an *event handler* and is defined as a *control flow.* However, event handler control flows are created on the Event Handlers Designer tab rather than on the Control Flow Designer tab.

SSIS

- The *Package Explorer* displays all the contents of the Integration Services package in a single tree-view structure. In addition to the control flow tasks, data flow tasks, and event handler tasks, the Package Explorer displays all the variables available for use in the package.

SSIS

- The Connections tray is used to manage all the connections in the package. This includes paths to flat files and connection strings for databases. Rather than having this spread throughout the package, all this connection information is stored in the Connections tray. This makes it much easier to locate and manage the connection information as paths, server names, and login credentials change, or as the package is used in different server environments.

SSIS

- As we create our Integration Services packages, we use a top-down approach. First, we use the Control Flow Designer tab to define the tasks that the package will perform. Next, we use precedence arrows to define the tasks' order of execution and any dependencies between them. Then, we drill down into each of the data flow tasks to define their operation. Finally, we add any event handlers or other programming needed to get the package functioning in the manner required.

SSIS

- We begin, then, on the Control Flow Designer tab, by specifying the tasks that must be accomplished by the package. We create tasks by dragging Control Flow items from the Toolbox and placing them on the Control Flow Designer tab. Each item taken from the Toolbox becomes a control flow task.

SSIS

- The red octagon with the white *X* on the control flow task indicates that the task is in an error state because some of the required properties for the task have not been set or have been set improperly. Hovering the mouse over a task with this symbol displays a message explaining the error. In other situations, a yellow triangle with a black exclamation point appears on a task. This indicates a warning message is associated with this task. Again, hovering the mouse over the task displays the warning message. A task with a warning executes, but a task with an error does not.

SSIS

- We can set the properties of a task by double-clicking the task to display its Editor dialog box. We can also select the task and modify its properties using the Properties window.

SSIS

- When creating an Integration Services package, it is important to change the default name for each task that is placed in the package. The names should be changed to a phrase that accurately describes the functionality performed by that task. It is also a good idea to enter a short explanation of each task in the Description property of the task. The description is displayed when we hover over the task on the Control Flow tab.

SSIS

- Any time a control flow task must make a connection to a database or to the file system, this connection information is stored in a Connection Manager in the Connections tray. This makes it easier to move packages from one server to another or from a development environment to a production environment. When database connection information or file system paths need to be modified to work on a different server, we do not need to look for them throughout the package. Instead, they are conveniently located in one place: the Connections tray.

SSIS

- Precedence Arrows
- We control the order in which tasks are executed by connecting the precedence arrow from one task to the task that is to run after it. To do this, we select a task and then click the green arrow that appears below that task. We drag this arrow until the mouse pointer is over the task that should follow during execution.

SSIS

- When we release the mouse button, the green arrow connects the two tasks, This is known as a *precedence constraint* because it constrains one task to run after another.

SSIS

- By default, the precedence arrow is green, meaning the second task will execute only after the successful completion of the first task. This is the “Success” precedence. When we right-click the arrow, a context menu appears, Three options on the context menu—Success, Failure, and Completion—allow us to select the type of precedence the arrow represents.

SSIS

- When we select Failure from the context menu, the precedence arrow changes from green to red. With “Failure” precedence, the second task executes only after the failure of the first task. When we select Completion from the context menu, the precedence arrow changes to blue. With “Completion” precedence, the second task executes after the first task has completed without regard to whether the first task succeeded or failed.

SSIS

- Data flow tasks are connected by flow path arrows. These are connected in the same way that control flow tasks are connected by precedence arrows. Green flow path arrows indicate the route that valid data will take through the task. This is the output flow of the task. Red flow path arrows indicate the route that invalid data follow. This is the error flow of the task.

SSIS

- The flow path from a data source to a data destination is called a *data flow segment.* The simplest data flow segment has only a single source and a single destination, connected by a flow path. This type of segment simply copies data from one place to another. Transformation tasks are added to the segment to modify the data as it moves from the source to the destination.

SSIS

- Data flow segments may contain multiple data sources, with the data flow from each source being combined with a merge or union task.

SSIS

- Likewise, a segment may have multiple data destinations, with the data flow being divided by an error path, a conditional split task, or a multicast task. We gain even more flexibility with the ability to have multiple, distinct data segments in the same data flow.

SSIS

- When a data flow has multiple segments, the order in which the segments are executed is determined by the execution plan created at run time. This means that we cannot depend on the order in which multiple segments are executed within a single data flow.

SSIS

- If the multiple segments are performing independent tasks, such as loading data into different dimensions of a data mart, this is not a problem. However, if these multiple segments are performing dependent tasks, such that one segment must be completed before another segment can begin, the segments should be split up into different data flow tasks. The order of execution can then be enforced by a precedence arrow from one data flow task to the other on the Control Flow Designer tab.

Control Flow

- The Control Flow Toolbox is divided into two areas: Control Flow items and Maintenance Plan tasks.

Control Flow

- Most components, Control Flow items, or Maintenance Plan tasks, have an editor dialog box that lets us configure that component in a straightforward manner. The editor dialog box is launched by double-clicking the component in the design area or right-clicking the item and selecting Edit from the Context menu.

Control Flow Items

- These items are presented in the order in which they appear in the Control Flow Toolbox. The Control Flow items are grouped into containers and tasks, with the containers at the top of the list. *Containers* are a special type of item that can hold other Control Flow items and Maintenance Plan tasks.

For Loop Container

- **For Loop Container** The *For Loop container* enables us to repeat a segment of a control flow. In this way, it functions much the same as a FOR... NEXT loop in Visual Basic or a *for* loop in C#.

Foreach Loop Container

- Like the For Loop container, the Foreach Loop container also provides a way of repeating a segment of a control flow. However, rather than have an expression to control when the loop is exited, the *Foreach Loop container* iterates one time for each item in a collection.

Sequence Container

- Unlike the For Loop and Foreach Loop containers, the Sequence container does not change the control flow. Instead, the purpose of the *Sequence container* is to help organize tasks in a package. The Sequence container can be used to do the following:

Sequence Container

- Organize a large package into logical sections for easier development and debugging.
- Manage properties for a group of tasks by setting the properties on the container, rather than on the individual tasks

Sequence Container

- Allow a group of tasks to be easily enabled or disabled to aid package development and debugging
- Provide a variable scope that includes multiple tasks, but does not include the entire package

ActiveX Script Task

- The *ActiveX Script task* enables us to define scripts using either VBScript or JScript. These scripts can be used to implement transformations or business logic that cannot be created using the other Integration Services tasks. The scripts can perform complex calculations and even use ActiveX COM objects to access database values or interact with the operating system.

ActiveX Script Task

- Using the ActiveX Script task in our Integration Services packages has three disadvantages. First, the ActiveX scripts are interpreted at run time. This can negatively impact performance and scalability. Second, the ActiveX scripts can be a security issue. A malicious script can use its direct access to server resources to do all kinds of bad things. Third, the ActiveX script editor does not provide any programmer aids, such as autocomplete, context-sensitive help, or debugging.

Bulk Insert Task

- The *Bulk Insert task* lets us rapidly copy data from a text file into a SQL Server table or view.

Data Mining Query Task

- The *Data Mining Query task* lets us execute a Data Mining Extensions (DMX) query against an existing data mining structure. The DMX query enables us to feed parameter values to a data mining model, and then have that mining model make predictions for us based on those parameters.

Execute Package Task

- The *Execute Package task* lets us execute a different Integration Services package. The package containing the Execute Package task is the parent package and the package being executed by that task is the child package. The child package can be stored in SQL Server or in a structured storage file.

Execute Process Task

- The *Execute Process task* enables us to execute a program or a batch file as part of an Integration Services package. This task can be used to do such things as unzipping compressed files.

Execute SQL Task

- The *Execute SQL task* enables us to execute SQL statements or stored procedures. The contents of variables can be used for input, output, or input/output parameters and the return value. We can also save the result set from the SQL statements or stored procedure in a package variable. This result set could be a single value, a multirow/multicolumn result set, or an XML document.

File System Task

- The *File System task* lets us perform one of the following file system functions:
- Copy a directory
- Copy a file
- Create a directory
- Delete a directory
- Delete the contents of a directory

FTP Task

- The *FTP task* enables us to perform the following functions on an FTP site:
- Send files
- Receive files
- Create a local directory
- Create a remote directory
- Remove a local directory
- Remove a remote directory
- Delete local files
- Delete remote files

FTP Task

- We must create an FTP Connection Manager in the Connections tray to specify the address and login credentials for the FTP server. In addition, a File Connection Manager or the content of a package variable is used to specify the local path for the transfer.

Send Mail Task

- The *Send Mail task* lets us send an e-mail message as part of our Integration Services package. This can be useful for alerting an administrator to an error condition or notifying a key user that a critical process has completed. The Send Mail task uses a Simple Mail Transfer Protocol (SMTP) server to deliver the e-mail. We must create an SMTP Connection Manager in the Connections tray to specify the SMTP server to use with this task.

XML Task

- The *XML task* enables us to manipulate XML documents. Using the XML task, we can perform the following operations:
- Validate an XML document using an XML Schema Document (XSD) or
- a Document Type Definition (DTD)
- Apply an XSL Transformation (XSLT)
- Apply an XPath query
- Merge two XML documents
- Find the difference between two XML documents (Diff operation)

SSIS

- DATA SOURCE
- DATA TRANSFORM
- DATA DESTINATION

Data Flow Transformations

- The *data flow transformations* are used to modify the data as it moves through the data flow. In most cases, the wide array of data flow transformations makes it possible to change the data into the required format without having to save it to a temporary table or utilize large amounts of custom code.

Aggregate

- The *Aggregate* transformation enables us to combine information from multiple records in the data flow into a single value. This functions in the same way as aggregation functions and the GROUP BY clause in a SQL statement.

Aggregate

- The following aggregations are available:
- Average
- Count
- Count distinct
- Maximum
- Minimum
- Sum

Cache Transform

- It enables us to populate a cache that will be subsequently used by a Lookup transformation. The Cache Transform can be configured to write the cached data to a cache file using a Cache Connection Manager. Only one Cache Transform in a package can write data to a given Cache Connection Manager.

Character Map

- The *Character Map* transformation enables us to modify the contents of character-based columns. The modified column can be placed in the data flow in place of the original column, or it can be added to the data flow as a new column.

Character Map

- **Lowercase** changes all characters to lowercase
- **Uppercase** changes all characters to uppercase
- **Byte reversal** reverses the byte order of each character
- **Hiragana** maps Katakana characters to Hiragana characters

Character Map

- **Katakana** maps Hiragana characters to Katakana characters
- **Half width** changes double-byte characters to single-byte characters
- **Full width** changes single-byte characters to double-byte characters

Character Map

- **Linguistic casing** applies linguistic casing rules instead of system casing rules
- **Simplified Chinese** maps traditional Chinese to simplified Chinese
- **Traditional Chinese** maps simplified Chinese to traditional Chinese

Conditional Split

- The *Conditional Split* transformation enables us to split the data flow into multiple outputs.

Copy Column

- The *Copy Column* transformation lets us create new columns in the data flow that are copies of existing columns.

Data Conversion

- The *Data Conversion* transformation enables us to convert columns from one data type to another. The converted data can either replace the existing column or be added as a new column.

Fuzzy Grouping

- The *Fuzzy Grouping* transformation enables us to find groups of rows in the data flow based on non-exact matches. This is most often used to find possible duplicate rows based on names, addresses, or some other column where the same information may have been entered in different ways.

Fuzzy Grouping

- For example, a row for Ms. Kathy Jones, a second row for Ms. Kathryn Jones, and a third row for Ms. Cathy Jones may be three entries for the same person.
- The Fuzzy Grouping transformation selects one of the rows in the group as the best candidate for the other rows to be combined into, and this is the *model row.*

Fuzzy Grouping

- Fuzzy Grouping transformation creates similarity scores between strings. This is done by considering the edit distance between two strings. In other words, how many character inserts, deletions, and replacements must be made in one string to produce the other string? Kathy and Kathryn have an edit distance of 2, because we need to insert an *r* and an *n* into the first word to get the second. Kathy and Cathy have an edit distance of 1, because we simply need to replace *K* with *C* in the first word to get the second.

Fuzzy Grouping

- In addition to edit distance, Fuzzy Grouping uses information, such as frequency of character occurrence and the positions of the characters, among other things, to increase its accuracy when creating similarity scores. Fuzzy Grouping is great at detecting character transpositions and other common spelling errors. All of the algorithms that create the similarity scores are language-independent, so Fuzzy Grouping can work with any language we care to throw at it.

Fuzzy Lookup

- The *Fuzzy Lookup* transformation lets us look up values using fuzzy matching logic. The Fuzzy Lookup transformation is closely related to the Fuzzy Grouping transformation. In fact, as was noted, the Fuzzy Grouping transformation creates its own lookup list, and then does a fuzzy lookup.

Fuzzy Lookup

- The Fuzzy Lookup transformation is extremely useful when the data is moving from a database where a column was entered as freeform text to another database where that same column uses a lookup table and a foreign key constraint.

Lookup

- The *Lookup* transformation works similarly to the Fuzzy Lookup transformation. The difference is the *Lookup transformation* requires exact matches, rather than using similarity scores.

Merge

- The *Merge* transformation merges two data flows together. For the Merge transformation to work properly, both input data flows must be sorted using the same sort order.

Merge

- This can be done by using the Sort transformation in each data flow prior to the Merge transformation. Alternately, if the rows coming from the data source are already sorted by the data provider, you can set the IsSorted property to true for the output of that data provider.

Merge

- The IsSorted property is only available in the Advanced Editor dialog box for the data source. The two data flows are merged together, so the output is sorted in the same manner as the inputs.

Term Extraction

- The *Term Extraction* transformation lets us extract a list of words and phrases from a column containing freeform text. The Term Extraction transformation identifies recurring nouns and/or noun phrases in the freeform text, along with a score showing the frequency of occurrence for each word or phrase. This information can then be used to help discover the content of unstructured, textual data.

References

- David Loshin, “Business Intelligence”, Morgan Kaufmann Publishers, 2003
- Mike Biere, “Business Intelligence for the Enterprise”, 2nd edition, IBM Press,2003.
- R N Prasad, Seema Acharya, “Fundamentals of Business Analytics”, Wiley India, 2011