What is SQL Server Integration Services (SSIS)?

**SQL Server Integration Services (SSIS)** is component of SQL Server 2005 and later versions. SSIS is an enterprise scale ETL (Extraction, Transformation and Load) tool which allows you to develop data integration and workflow solutions.

* Can be used to accomplish a broad range of data migration tasks.

Important **components**:

* Data flow
* Control flow
* Package explorer
* Event handler

**SSIS package** => an organized collection of connections like data flow elements, control events, event handlers, parameters, variables, and configurations.

You assemble them by either building it programmatically or by graphical design tools that SSIS provides.

**Types of connections or files** that work within SSIS are

* ODBC
* OLEDB
* .net SQLClient
* Flat File
* Excel
* XML

What are SSIS Connection Managers?

When we talk of integrating data, we are actually pulling data from different sources and writing it to a destination. But how do you get connected to the source and destination systems? This is where the connection managers come into the picture. **Connection manager** represent a connection to a system which includes data provider information, the server name, database name, authentication mechanism, etc.

What is the **RetainSameConnection** property (of Connection Manager) impact?

Whenever a task uses a connection manager to connect to source or destination database, a connection is opened and closed with the execution of that task. Sometimes you might need to open a connection, execute multiple tasks and close it at the end of the execution. This is where RetainSameConnection property of the connection manager might help you. When you set this property to **TRUE**, the connection will be opened on first time it is used and **remain open until execution of the package completes**.

What are a source and destination adapters?

**A source adaptor** basically indicates **a source in Data Flo**w to pull data from. The source adapter uses a connection manager to connect to a source and along with it you can also specify the query method and query to pull data from the source.

Similar to a source adaptor, the destination adapter indicates **a destination in the Data Flow** to write data to. Again like the source adapter, the destination adapter also uses a connection manager to connect to a target system and along with that you also specify the target table and writing mode, i.e. write one row at a time or do a bulk insert as well as several other properties.

Please note, the source and destination adapters can both use the same connection manager if you are reading and writing to the same database.

How does SSIS differ from DTS (**SSIS vs DTS**)?

SSIS is a successor to **DTS (Data Transformation Services)** and has been completely re-written from scratch to overcome the limitations of DTS which was available in SQL Server 2000 and earlier versions. A significant improvement is the segregation of the control/work flow from the data flow and the ability to use a **buffer/memory oriented architecture** for data flows and transformations which improve performance.

**Control Flow** => allows you to order the workflow, so you can ensure tasks/components get executed in the appropriate order.

What is the **Data Flow Engine** (**SSIS pipeline engine**)?

* Responsible for managing the flow of data from the source to the destination and performing transformations (lookups, data cleansing etc.).

Data flow uses **memory oriented architecture**, called buffers, during the data flow and transformations which allows it to execute extremely fast. This means the SSIS pipeline engine pulls data from the source, stores it in buffers (in-memory), does the requested transformations in the buffers and writes to the destination.

The benefit is that it provides the fastest transformation as it happens in memory and we don't need to stage the data for transformations in most cases.

**Transformation** => Bringing in the data in a desired format.

Ex. - 1. Ensure only distinct records are written to the destination, so duplicates are removed. 2. If you have master/reference data and want to pull only related data from the source and hence you need some sort of lookup.

What is a Task?

A **task** is very much like a method of any programming language which represents or carries out an **individual unit of work**.

Broadly **two categories** of tasks in SSIS

* **Control Flow tasks**: operational in nature except Data Flow tasks.
* **Database Maintenance tasks**.

What is a **Precedence Constraint** and what types of Precedence Constraint are there?

SSIS allows you to place as many as tasks you want to be placed in control flow. You can connect all these tasks using **connectors** called **Precedence Constraints**.

* Allow you to define the logical sequence of tasks in the order they should be executed.
* Can also specify a condition (either a constraint, an expression or both) to be evaluated before the next task in the flow is executed.

Types of precedence constraints and the condition could be:

* Success (next task will be executed only when the last task completed successfully) or
* Failure (next task will be executed only when the last task failed) or
* Complete (next task will be executed no matter the last task was completed or failed).

What is the **Data Path** and how is it different from a Precedence Constraint?

**Data Path** is used in a Data Flow task to connect to different components of a Data Flow and show transition of the data from one component to another. A data path **contains the meta information** of the data flowing through it, such as the columns, data type, size, etc.

**Data path Vs precedence constraint:** the data path is used in the data flow, which shows the flow of data whereas the precedence constraint is used in control flow, which shows control flow or transition from one task to another task.

What is a Data Viewer utility and what it is used for?

The **data viewer utility** is **used** in Business Intelligence Development Studio during development or **when troubleshooting an SSIS Package**. The data viewer utility is placed on a data path to see what data is flowing through that specific data path during execution. The data viewer utility displays rows from a single buffer at a time, so you can click on the next or previous icons to go forward and backward to display data.

What is an SSIS breakpoint? How do you configure it? How do you disable or delete it?

A **breakpoint** allows you to **pause the execution** of the package in Business Intelligence Development Studio during development or **when troubleshooting an SSIS Package**. You can right click on the task in control flow, click on **Edit Breakpoint menu** and from **the Set Breakpoint** window, you specify when you want execution to be halted/paused. For example OnPreExecute, OnPostExecute, OnError events, etc. To toggle a breakpoint, delete all breakpoints and disable all breakpoints go to the **Debug menu** and click on the respective menu item.

**SSIS event logging**: Like any other modern programming language, SSIS also raises different events during package execution life cycle. You can enable or write these events to trace the execution of your SSIS package and its tasks. You can also write your custom message as a custom log. You can enable event logging at the package level as well as at the tasks level. You can also choose any specific event of a task or a package to be logged. This is essential when you are troubleshooting your package and trying to understand a performance problem or root cause of a failure.

**SSIS log providers:** There are several places where you can log execution data generated by an SSIS event log:

1. SSIS log provider for Text files
2. SSIS log provider for Windows Event Log
3. SSIS log provider for XML files
4. SSIS log provider for SQL Profiler
5. SSIS log provider for SQL Server, which writes the data to the msdb..sysdtslog90 or msdb..sysssislog table depending on the SQL Server version.

**Event handling in SSIS:** You can write an even handler to capture the event and handle it in a few different ways. For example consider you have a data flow task and before execution of this data flow task (**OnPreExecute** event) you want to make some environmental changes such as creating a table to write data into, deleting/truncating a table you want to write, etc. Along the same lines, after execution of the data flow task (**OnPostExecute** event) you want to cleanup some staging tables.

What is a **container** and how many **types of containers** are there?

A **container** is a logical grouping of tasks which allows you to manage the scope of the tasks together.

**Types:**

1. **Sequence** Container - Used for grouping logically related tasks together
2. **For Loop** Container - Used when you want to have repeating flow in package
3. **For Each Loop** Container - Used for enumerating each object in a collection; for example a record set or a list of files.

Apart from the above mentioned containers, there is one more container called the **Task Host Container** which is not visible from the IDE, but every task is contained in it (the default container for all the tasks).

What are variables and what is variable scope?

A **variable** is used to store values.

Two Types:

**System Variable** (like ErrorCode, ErrorDescription, PackageName etc) whose values you can use but cannot change and

**User Variable** which you create, assign values and read as needed.

A variable can hold a value of the data type you have chosen when you defined the variable.

Variables can have a different scope depending on where it was defined. For example you can have package level variables which are accessible to all the tasks in the package and there could also be container level variables which are accessible only to those tasks that are within the container.

What is the **transaction support** feature in SSIS?

When you execute a package, every task of the package executes in its own transaction. What if you want to execute two or more tasks in a single transaction? You can group all your logically related tasks in single group. Next you can set the transaction property appropriately to enable a transaction so that all the tasks of the package run in a single transaction. This way you can ensure either **all of the tasks** **complete** **successfully** **or** if any of them fails, the transaction gets **roll-backed too.**

**ODBC vs OLEDB**

Those are just different APIs for the different data sources (i.e. databases). OLE DB is newer and arguably better until 2008. MS have deprecated OLEDB and SQL Server no longer supports it (SS 2012 was the last to support it).

OLEDB which was introduced in 1998 was meant to be a replacement for ODBC (introduced in 1992).

ODBC (Open Database Connectivity) is designed to provide access primarily to SQL data in a multi-platform environment. OLE DB (Object Linking and Embedding Database) is designed to provide access to all types of data in an OLE Component Object Model (COM) environment.

ODBC => for relational databases (SQL Server, Oracle etc).

OLE DB: => for both relational and non-relational databases. (Oracle, SQL Server, Excel, raw files, etc).

Also, there exists OLEDB provider (wrapper) for ODBC which allows one to use OLEDB to access ODBC data sources if one so wishes.

**Vs JDBC**

**ODBC** is an open interface which can be used by any application to communicate with any database system, while JDBC is an interface that can be used by Java applications to access databases. Therefore, unlike JDBC, ODBC is language independent.