**Types of fields in a Dimension**

■ Keys - Used to identify entities

■ Name columns - Used for human names of entities

■ Attributes - Used for pivoting in analyses. Continuous values should be discretized according to business rules.

■ Member properties - Used for labels in a report such as company name, address etc. Not pivoted on

■ Lineage columns - Used for auditing, and never exposed to end users

**Hierarchies**

Dimension can have natural hierarchies that provide drill-down path. Hierarchies have levels which have members eg Jan, Feb, March etc.

In a snowflake schema, hierarchies are implemented as joined look-up tables.

In a star schema, they are implemented in terms of attributes

**Slowly Changing Dimensions**

Type 1 SCD – update the single row. Used when there is no concept of history for that attribute eg product name is changed there is no sense analysing the old name

Type 2 SCD – create a new row. Used when there is a concept of history eg someone moves from one address to another. If doing this then need a surrogate key because there will be duplicates.

**Types of fields in a Fact table**

Measure – normally numerical, can be aggregated and what is being measured eg sales value.

Foreign keys to dimensions. – all foreign keys together usually uniquely identify a row.

Linage

**Additivity of Measures**

Additive – measure can be summed across any dimension eg sales amount

Non-Additive – can’t be summed across any dimension. Normally averaged eg unit price

Semi-Additive – can be summed across any dimension except time. Eg balance

**Related fact tables**

Fact tables may be related and so one may be on ‘many’ side. This won’t work for SSAS so can put in an intermediary Dimension table. Not derived from source, is a construct of the data warehouse.

**Calculating space needed by DW**

For data files, extrapolate from a small period and add 25% for rebuilding indexes without fragmentation.

For log, find your biggest transaction

**Files Groups –** have one group per partition. Read this when have a chance

https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee796978(v=cs.20)

**DW schemas** – one for ETL one for the DW

**Sequence** – are independently created from a table and can be used by multiple tables. Can perform better and offer more functionality

**Computed columns** – can be used to discretized columns

**Fact tables – foreign keys** – can be disabled for loading into but ETL should be robust enough to error if violations occur. Or it can add a row when it doesn’t exist. This is called **inferred member**.

**Indexes**

Clustered Index, why?

Can defrag the table

Move table to a new file groups without having to drop and recreate

By default, a non-clustered index will be created which will probably be bigger than a clustered index.

Can create covered queries that run only off the index.

Opt for an integer autonumbering surrogate key as the clustered primary key for all DW tables, unless there is a really strong reason to decide otherwise.

Non-clustered index, won’t be used much on dimension attributes. Could be used on name columns and member property columns, used in parametrised reports eg company name.

Probably don’t need indexes on the foreign keys, analyse type of join being done eg is it hash or merge?

**Indexed Views**

Optimize queries that aggregate data and perform multiple joins. They basically material a view and will be used if a query

**Compression** – useful when data is mostly read and rarely updated

Row compression…

*Row compression*reduces metadata overhead by storing fixed data type columns in a variable-length format. This includes strings and numeric data. Row compression has only a small impact on CPU resources and is often appropriate for OLTP applications as well.

Page compression…

*Page compression*includes row compression, but also adds prefix and dictionary compressions. *Prefix compression*stores repeated prefixes of values from a single column in a special compression information (CI) structure that immediately follows the page header, replacing the repeated prefix values with a reference to the corresponding prefix.*Dictionary compression*stores repeated values anywhere in a page in the CI area. Dictionary compression is not restricted to a single column.

Unicode compression – stores Unicode strings as single bytes where able to

**Columnstore index**

Stores indexes by column rather than row. Not good for very selective queries.  Not suitable for OLTP workloads. And force table to be read-only. Must drop and recreate. One columnstore index per table. So

Use columnstore indexes for

■■Read-mostly workloads.

■■Updates that append new data rather then add.

■■Workflows that permit partitioning or index drop/rebuild.

■■Queries that often scan and aggregate lots of data.

Don’t use columnstore indexes when

■■You update the data frequently.

■■Partition switching or rebuilding indexes doesn’t fit your workflow.

■■Your workload includes mostly small lookup queries.

**Batch mode processing**

Retrieves data in batches rather than individual rows. Lowers CPU. Useful for bitmap filtered hash join and scan operators

**Partitioning**

Tables and indexes can be partitioned. Often done on dates.

Partition function - An object that maps rows to partitions by using values from specific columns (partitioned columns)

Partition scheme – Maps partitions to filegroups

Aligned index – indexes that have the same portioning as base table. Columnstore indexes **have** to be aligned with their base tables

Partition elimination – query optimizer only access partitions it needs to

Partition switching - This is a process that switches a block of data from one table or partition to another table or partition.

If you want to switch content from a nonpartitioned table to a partition of a partitioned table, what conditions must the nonpartitioned table meet?

* It must have the same constraints as the partitioned table.
* It must have the same compression as the partitioned table.
* It must have a check constraint on the partitioning column that guarantees that all of the data goes to exactly one partition of the partitioned table.
* It must have the same indexes as the partitioned table.

**Data Linage**

Many in-built functions in SSIS and T-SQL for logging linage eg

APP\_NAME() - ApplicationName

DATABASE\_PRINCIPAL\_ID() DatabasePrincipalId

USER\_NAME() DatabasePrincipalName

SUSER\_SID() ServerPrincipalId

SUSER\_SNAME() ServerPrincipalName

CONNECTIONPROPERTY('net\_transport') transport protocol

CONNECTIONPROPERTY('client\_net\_address') Client net address

CURRENT\_TIMESTAMP

@@ROWCOUNT – number of rows process

**Input and Export Wizard**

Use input/export wizard for simple data transfer ie where data is not being transformed

**SSIS packages are XML**

**SSIS parameterization replace SSIS configuration**

**Things SSIS can do**

Access remote locations eg FTP

Call processes that are external to SSIS

SQL Server Administration operations eg backups, integrity checks

Operating system inspection - Windows Management Instrumentation (WMI) data is accessible to SSIS

Send mail

SQL Server Analysis Services processing

Data profiling

Data mining queries

**Connection Manager Types**

ADO connection manager – connect to active X. Used for backward compatibility so don’t use

ADO.NET connection manager – connects to data stores using .net. Can be used for SQL Server

Analysis Services connection manager – connects to SSAS database

File connection manager/Multiple Files connection manager – connects to SSIS data files

Flat File connection manager/Multiple Flat Files connection manager – connects to flat files

FTP connection manager – SFTP not supported and windows integrated authentication not supported

HTTP connection manager – connects to web service. Again basic authentication is supported not windows integrated

MSMQ connection manager - access to Microsoft Message Queuing (MSMQ) message queues.

ODBC connection manager – being deprecated

OLE DB connection manager - access to database management systems that use the OLE DB provider include SQL Server

SMO connection manager - access to SQL Management Object (SMO) servers. Used for maintenance tasks

SMTP connection manager - The SMTP connection manager provides access to Simple Mail Transfer Protocol (SMTP) servers and is used by the Send Mail task to send email messages.

SQL Server Compact Edition connection manager – used to access SQL Server Compact Edition (whatever that is)

WMI connection manager - connect to Windows Management Instrumentation ie OS stuff

**What to use**

ADO should be used when using parametrized queries in Execute SQL task because

- can use name of parameter rather than question mark

-Supports additional datatypes eg VARCHAR(MAX) and VARBINARY(MAX) and Binary

**Project level connection manager** – used by all packages in a project. If have same name then package level takes precedents. If you convert a project level connection to package connection then other packages won’t be able to use it.

**At design time**, connection managers are used by the SSIS developer to configure a connection to a data source.

**At run time**, connection managers are used by the SSIS engine to establish connections to data sources.

THINK HOW TO implement parameterization of sensitive info eg passwords

**Complex Data Movement** has these features

Data cleansing -

Data normalization - conversion of complex data types into primitive data types eg by parsing XML

Data type conversion –

Data translation - eg changing F to Female

Data validation - This is the verification and/or application of business rules against individual values

Data calculation and data aggregation - for example, “net price” and “tax” exist at the source, but “price including tax” is expected at the destination

Data pivoting and data unpivoting - Source data might need to be restructured or reorganized in order to comply with the destination data model (for example, data in the entry-attribute-value (EAV) might need to be restructured into columns or viceversa).

**Data Preparation Tasks**

File System task – operations on task eg copying moving etc.

FTP task – typically used to download from or upload to an FTP.

Web Service task – access web services

XML task – XML manipulation, validation and data retrieval

Data Profiling task – used for determining data quality and cleansing

**Workflow Tasks**

Execute Package task – call another SSIS package

Execute Process task – execute process outside SQL

Message Queue task - used to send and receive messages to and from Microsoft Message Queuing (MSMQ) queues on the local server.

Send Mail task - allows the sending of email messages from SSIS packages by using the Simple Mail Transfer Protocol (SMTP).

WMI Data Reader task - provides access to Windows Management Instrumentation (WMI) data which gives access to information about the environment

WMI Event Watcher task - Typically, the WMI Event Watcher task would be used to trace events in the environment, and based on them to control the execution of SSIS processes

Expression Task - Typically, the Expression task is used to assign values to variables without the overhead of using the Script task for the same purpose.

CDC Control task - This task controls the life cycle of SSIS packages that rely on the SQL Server 2012 Change Data Capture (CDC) functionality.

**Data Movement Tasks**

Bulk Insert task – Task loads data from text files to SQL server database tables. No transformation options so is loaded quickly. Requires Sysadmin rights.

Execute SQL task –

Data flow task – allows complex data movements

**SQL Server Administration Tasks**

Transfer Database task – online – slower but database is available during backup, offline – faster but database is not available

Transfer Error Messages task - Use this task to transfer user-defined error messages from one SQL Server instance to another;

Transfer Jobs task - Use this task to transfer SQL Server Agent Jobs from one SQL Server instance to another

Transfer Logins task - Use this task to transfer SQL Server logins from one SQL Server instance to another

Transfer Master Stored Procedures task - Use this task to transfer user-definedstored procedures (owned by dbo) from the master database of one SQL Server instance to the master database on another SQL Server

Transfer SQL Server Objects task - Use this task to transfer objects from one SQL Server instance to another

**SQL Server Maintenance Tasks**

Back Up Database task –

Check Database Integrity task - automate data and index page integrity checks

Execute SQL Server Agent Job task

Execute T-SQLStatement task – more basic version of “Execute SQL task”, can’t send parameters for eg

History Cleanup task - Use this task in your maintenance plan to automate the purging of historical data about backups and restore operations

Maintenance Cleanup task - Use this task in your maintenance plan to automate the removal of files left over by maintenance plan execution

Notify Operator task - Use this task in your maintenance plan to send email messages to SQL Server Agent operators.

Rebuild Index task

Reorganize Index task

Shrink Database task – avoid shrink because might cause fragmentation

Update Statistics task

**Analysis Services Tasks**

Analysis Services Execute DDL task - This task provides access to SSAS databases for creating, modifying, and deleting multidimensional objects or data mining models.

Analysis Services Processing task - This task provides access to SSAS databases to process multidimensional objects, tabular models, or data mining models.

Data Mining Query task - This task provides access to Data Mining models, using queries to retrieve the data from the mining model and load it into a table in the destination relational database

**The Script Task**

Used to provide functionality not available from the inbuild features. Avoid using unless have to.

**Custom Tasks**

Use instead of script task if reusability is required eg different SSIS packages uses same script. Can be developed independently of SSIS.

**Containers**

Provide structure, encapsulation (eg set of tasks can be executed repeatedly) and scope.

For-Loop Container – executes control flow until evaluation condition is false eg using a variable.

Foreach Loop Container – executes control flow over an enumerate eg all files in a folder or number of rows in a table. Assign enumberater value to a variable to make available to tasks inside the loop

Sequence Container – controls structure and determine precedents.

**Data Flow Task**

At run time, the data flow task builds an execution plan from the data flow, and the data flow engine executes the plan.

**Source Assistant -** The Source Assistant helps you create a source adapter and connection manager.

**ValidateExternalMetadata** – turn off if the object doesn’t exist yet eg is being created by the package

**Data flow destination**

ADO.NET destination

Data Mining Model Training - Allows you to pass data from the data flow into a data mining model in SSAS.

DataReader destination - Lets you pass data in a ADO.NET recordset that can be programmatically referenced.

Dimension Processing - Loads and processes a SSAS dimension.

Excel destination

Flat File destination

ODBC destination – allows batch mode

OLE DB destination

Partition Processing - Allows an SSAS partition to be processed

Raw File destination - Stores data in native SSIS format as a binary file.

Recordset destination

SQL Server Compact destination

SQL Server destination - Provides a high-speed destination specific to a local SQL Server database.

**Fast Parse**

Very fast loading of flat files only works on certain data types

**Support for delimited files with varying numbers of columns** eg ragged-right

**Things to remember**

■■Use appropriate data source or data destination adapters.

■■ Always extract only the columns you need.

■■ Use Fast Load or Batch mode when inserting data by using an ODBC or OLE DB destination

adapter.

■■ Use a Raw File destination if you have to temporarily store data to be used by SSIS

later.

**Blocking in transformations**

non-blocking transformations – each row goes through transformation without waiting

partial-blocking transformation – waits until a sufficient number of rows have been stored then proceeds

blocking transformation – all rows must read into the transformation before they are be processed

**Logical Row-Level Transformations**

Performs operations at row level – use eg calculated columns from multiple sources and conversion of data types

Audit - Adds additional columns to each row based on system package variables such as ExecutionStartTime and PackageName.

Cache Transform - Allows you to write data to a cache with the Cache connection manager. The data can then be used by the Lookup transformation. This is useful if you are using multiple Lookup transformations against the same data, because SSIS will cache the needed data only once and not for each Lookup component.

Character Map - Performs common text operations such as Uppercase and allows advanced linguistic bit-conversion operations.

Copy Column

Data Conversion

Data Conversion

Export Column - Exports binary large objects (BLOB) columns, one row at a

time, to a file.

Import Column - Loads binary files such as images into the pipeline; intended

for a BLOB data type destination.

Row Count - Tracks the number of rows that flow through the transformation and stores the number in a package variable after the final row.

**Useful expressions**

REPLACENULL (col1, 0) – If Col1 is null then replace with 0

NULL(DT\_I4) – puts a null into a 4 byte integer column

**Multi-Input and Multi-Output Transformations**

CDC Splitter

Conditional Split – rereoutes rows according to condition

Lookup - Performs a lookup operation between a current row and an external dataset on one or more columns. Used in Data Warehouses to get the foreign key.

Merge - Combines the rows of two similar sorted inputs, one on top of the other, based on a defined sort key. – partially blocking

Merge Join - Joins the rows of two sorted inputs based on a defined join column or columns, adding columns from each source. Works like T-SQL join but each source has to be sorted on joining columns– partially blocking

Multicast - Generates one or mode identical outputs, from which every row is sent out every output. This transformation creates a logical copy of the data.

Union all – Partially blocking

**Multi-Row Transformations**

Aggregate – Same as in SQL eg AVG, SUM etc

Percent Sampling – allows only defined number to go through

Pivot

Row Sampling - Generates a fixed number of rows, sampling the data from the entire input, no matter how much larger than the defined output the input is.

Sort – used for merge joins and for removing duplicates

Unpivot

Sort, Aggreate and Row Sampling are fully blocked so may be memory intensive

**Advanced Data-Preparation Transformations**

DQS Cleansing - Validates rows by automatically performing data cleansing using an existing knowledge base in Data Quality Services (DQS).

OLE DB Command - Performs database operations such as updates or deletions, one row at a time, based on mapped parameters from input rows.

Slowly Changing Dimension

Data Mining Query – Applies input rows against a data mining model

Fuzzy Grouping – Performs de-duplication based on similarity of selected columns – full blocking

Fuzzy Lookup - Joins a data flow input to a reference table based on column similarity. The Similarity Threshold setting specifies the closeness of allowed matches. – full blocking

Script Component –

Term Extraction - Analyzes text input columns for English-language nouns and noun phrases.

Term Lookup - Analyzes text input columns against a user-defined set of words for association.

**Resolve References editor**

use to quickly resolve the mapping of input and output columns between components.

**ETL Architecture**

ETL solution using SSIS should have multiple packages dedicated to each process. Means it is more modular and can be better developed

**Lookup Transformation**

Full cache – database queried once and start of execution of the package and results cached. But you need enough memory. Will NOT swap memory so if not enough memory will fail.

Partial cache – requeried at the beginning of the data flow task. When a new row comes in, the Lookup transformation checks its cache for the matching values. If no match is found, it queries the database. If the match is found at the database, the values are cached so they can be used the next time a matching row comes in.

No cache – stores only the last row

Because the whole table is being put in memory, write SQL so only the required columns are included

Lookup transformation are case sensitive. Can you Character Map transformation to convert case

Best approach for missing lookup – (if you want the rows) set to “ignore failure” then add derived column task to replace eg ISNULL(CustomerDwKey) ? 0 : CustomerDwKey.

Lookup transformation doesn’t need to be sorted. Much more efficient than merge joins.

**Cache Transform transformation**

The Cache Transform transformation writes data from a connected data source in the data flow to a Cache connection manager which persists in memory whilst package is running (or whole project if set at that level.) Or persisted to disk. Benefits:

* Query the lookup once and used for multiple look ups eg for **role-playing dimension** (dimensions used for multiple looks eg Date)
* You can do lookups against other (non OLE-DB) sources.

**Sorting**

Is expensive in SSIS so can be done in SQL using custom SQL then show the source is sorted with SortKeyPosition and IsSorted

**Set-Based Updates**

Don’t do in SSIS. Push back to SQL

**Parameters v Variables** – parameters are exposed to the caller, variables are not. And parameters are read-only. Parameters collects values that should be determined outside SSIS process.

**Variables**

Can store row sets but if large may cause package to run out of memory.

Some of the data types…

Object – general type not represented by another type

Byte - integers with values between 0 and 255.

DBNull – used to assign an explicit null

Int16 - 16-bit integers with values between –32,768 and 32,767

Int32 - signed 32-bit integers with values between –2,147,483,648 and 2,147,483,647

Single – Floating point with precision of seven digits

Decimal –

String – Unicode characters

Two types of variables can store row sets. If rowset comes from results of query then Object. If rowset is XML can also be stored in a string and handled Data Flow XML

Source Component.

Namespace – two User and System

**Property Parameterization** allowing specific SSIS object properties to be set dynamically, can be implemented in several ways:

Set from calling environment

Explicit assignment of a property from a variable eg SQL from Variable

Assignment through expressions

**What should be typically parameterized?**

Connection managers

Tasks and components – if component relies on values determined by additional programmatic logic

Data flow tasks - Large data movements are typically resource intensive; therefore, in order to prevent them from running out of resources, you could adjust their behavior in accordance with the actual availability of resources at run time by using appropriate programmatic logic eg setting batch size

SSIS expression **not** part of the .net framework

**Execute Package task**

Two methods for parameterizing child package.

Package configuration - For every property that should be exposed to the caller (the parent package), a parent package variable configuration must be prepared in the child package. The name of the parent package variable must match the name of the corresponding variable in the parent package.

Parameters - Variables, project parameters, or package parameters of the parent

package can be mapped to the parameters of the child package belonging to the

same project as the master package. This method is only available in the Project Deployment

model.

**Inferred Dimension Members**

When a fact arrives with a business key not in the dimension. Deal with as follows:

* Create a new dimension record with the business key (flag it in some way)
* Create the fact record
* Don’t treat inferred dimension as type 2 SDT because, when it does come in, it will generate new records. Instead, when it comes in, update it and remove flag.

**Slowly Changing Dimension**

Easy to use but only use for dimensions up to 10000 rows

**hash function** - algorithm or subroutine that maps large data sets of variable length, called keys, to smaller data sets of a fixed length. Useful for performing deltas.

**Passing dynamic SQL to source**

Can be done with a parameterized SQL command

Build SQL command in an expression of a variable

ODBC or ADO Net can’t accept parameters for their SQL so have to done via their assigning an expression to the SQL command property.

Package level parameters

**change data capture** – can be set up on SQL database on a table-by-table basis that will identify records that have changed. SSIS can then pick this up using CDC control. The CDC source offers five possibilities of supplying data:

All – row will be supplied for every change. So if record gets updated 3 times, 3 rows will be supplied

All with old values – as All but two rows for every change, one before, one after.

Net – One row per unique row. Best for DW.

Net with update mask – as Net with additional Boolean column for each column updated

Net with merge – as Net but no distinction between update and insert

**Loading Fact tables**

Partition fact table with incremental data on one partition so can be easily switched.

Use this strategy:

* Load incremental data to a table that has the same structure as the destination fact table, without compression or indexes.
* Apply the necessary indexes and compression.
* Switch the loaded table with the partition in the destination fact table.

Use fully cached lookups to get appropriate surrogate keys.

**Error Flows**

Three options for handling errors in the data flow components

* Fail Transformation – whole transfer fails
* Ignore Failure – value is nullified and transfer continues
* Redirect Rows – sends the row down the red path

Individual columns can have different settings.

**Transactions**

Transactions in SSIS use the Microsoft Distributed Transaction Coordinator, allows distributed transactions providing it has been turned on. Eg could work over SQL and Oracle database.

To enable transactions in SSIS, you must start the MSDTC service, and the tasks that

you want to include as part of the transaction must work with MSDTC services natively.

To enable transactions set TransactionOption. This exists at package, container and control flow task level.

Can be set to the following:

Required - If a transaction already exists, join it; if not, start a new transaction.

Supported - If a transaction exists, join it (this is the default setting).

NotSupported - The package, container, or task should not join an existing

transaction.

If a series of tasks must be completed as a single unit in which either all the tasks are

successful and committed or an error occurs and none of the tasks are committed,

place the tasks within a sequence container and set the TransactionOption property of

the container to Required.

A task can inherit the transaction setting of its parent when the TransactionOption

property is set to Supported,

If you want to exclude a specific task from participating in a transaction, set the

TransactionOption property to NotSupported.

If you set the TransactionOption property of a Foreach Loop container or For Loop

container to Required, a new transaction will be created for each loop.

Running tasks in transactions has a performance overhead

Configuring the MSDTC across multiple hosts and environments in complex and sometimes fragile

**Transaction Isolation Levels**

Unspecified - A different isolation level than the one specified is being used, but the level cannot be determined.

ReadUncommitted – Does not lock the record being read ie dirty read

Chaos - Same as ReadUncommitted, but checks the isolation level

of other pending transactions during a write operation so that transactions with more

restrictive isolation levels are not overwritten.

ReadCommitted – Looks the record being read but then immediately releases it.

RepeatableRead - Locks the records being read and keeps the lock until the transaction

completes.

Serializable - Locks the entire data set being read and keeps the lock until the transaction

completes.

Snapshot - The data read within a transaction will not reflect changes made by other

simultaneous transactions. The transaction uses the data row versions that exist when

the transaction begins. No locks are placed on the data when it is read.

RetainSameConnection – connection manager will join existing connection rather than start a new one.

**Checkpoints**

CheckpointUsage = Always – checkpoint file will have to exist or the package won’t run. Also using checkpoints is not allowed if you have set the TransactionOption of the package to Required.

**Event Handlers**

|  |  |
| --- | --- |
| OnError | Runs when an executable component reports an error |
| OnExecStatusChanged | Runs for all tasks and containers when the execution status changes  to In Process, Success, or Failed |
| OnInformation | Runs when SSIS displays information messages during the validation and execution of a task or container |
| OnPostExecute | Runs after a container or task successfully completes |
| OnPostValidate | Executes after the task or container has been successfully validated |
| OnPreExecute | Runs before an executable component is executed |
| OnPreValidate | Runs before a component is validated by the engine |
| OnProgress | Executed when a progress message is sent by the SSIS engine, indicating tangible advancement of the task or container |
| OnQueryCancel | Invoked when an Execute SQL task is canceled through manual intervention, such as stopping the package |
| OnTaskFailed | Similar to OnError, but runs when a task fails rather than each time an error occurs |
| OnVariableValueChanged | Runs when the value changes in a variable for which the RaiseChangeEvent property is set to True |
| OnWarning | Runs when a task returns a warning event such as a column not being used in a data flow |

Event Handlers can be turned off for a particular task with DisableEventHandlers property

**Parameters**

Package and project level

Cannot be changed whilst package is executing

Parameter design values are stored in the project file.

**Package Configuration**

To use package configuration, have to be in package deployment mode. By default package is in project deployment mode.

Configuration types:

* XML file
* Environment variable
* Registry Entry
* Parent Package Variable – can be used to send a variable from a parent package to its child
* SQL Server

indirect file location approach – using a environment variable as the file location pointer. Useful when the location of the file will change from one environment to another.

**Auditing**

Integration Services Logging – logs information about each execution

Integration Services Auditing – data flow transformation

**Logging packages**

Five log providers

* Text file
* SQL Server Profiler – writes log to a trace
* SQL Server - Writes log entries to the sysssislog system
* Windows Event Log – Writes logs to Application log that can be viewed Windows Event Viewer
* XML file

**How will log be used?**

* Low volume and only need for error detection – consider application log or text file
* Need for performance monitoring – SQL server or profiler

Should implement more than one logging, for example, SQL server and Text file in case SQL itself goes down.

**Determining what events need to be logged**

* execution boundary events – start and end of execution task
* execution progress events – eg variable changes value
* execution exception

exception and boundary events needed for high-level monitoring

**Event properties that can be captured**

Computer – name of computer event occurred

Operator –

SourceName – Name of container or task event occurred in

SourceID – identifier of package, container

ExecutionID – GUID of the execution instance

MessageText – most useful for trouble-shooting

StartTime

EndTime

DataCode –

0 – Success

1 – Failure

2 – Completed

3 – Canceled

**LoggingMode** – UseParentSetting = logging settings will be same as parent object

Log Configuration Templates – logging setting can be saved as template to use in other pacakges.

**Auditing**

Can be divided into 2 types:

Elementary auditing – one-to-one with the record being audited. Changes in data, who made it, when etc. Usually stored in same place at the aduit eg database

Complete auditing – many-to-one with the record being audited eg before and after values. Usually stored in different location

Auditing is less important in DW than in OLTP

**Package Templates**

**Security Considerations**

Virtual accounts - Virtual Accounts can access the network by using the computer identity in a domain environment.

Managed service accounts - Functionally equivalent to other domain accounts, even though they only exist on the local server. Typically this is used for an SSIS service

Domain account

**Upgrade considerations**

Before upgrading SSIS - you should be familiar with the limitations regarding versions and editions of SQL Server, as well as whether the target environment is a 32-bit or 64-bit environment.

**SSIS Tools**

|  |  |
| --- | --- |
| SQL Server Import And Export Wizard | Used to copy data between supported data stores |
| SQL Server Integration Services Deployment Wizard Used | Used to deploy SSIS projects to an instance of SQL Server. |
| SQL Server Integration Services Project Conversion Wizard Used | Used to generate a project deployment file from a set of SSIS package files and accompanying configuration files. |
| SQ L Server Integration Services Package Upgrade Wizard | Used to upgrade SSIS packages created in previous versions of SQL Server |
| SQL Server Integration Services Package Installation Utility | Used to deploy SSIS packages to an instance of SQL Server using the deployment manifest. Legacy reasons |
| SQL Server Integration Services Package Utility | Used to manage SSIS packages (for example, to copy, move, or delete them, or to verify their existence) from the command line. |
| SQL Server Integration Services Execute Package Utilities | Two utilities used to execute SSIS packages, either from the command line or from a user interface. |

**What are SQL Server Integration Services (SSIS)?**

SSIS is a feature of SQL Server that hosts SSIS deployment, maintenance, execution, and monitoring.

**How can SSIS be installed?**

SSIS can be installed together with other SQL Server features, added to an existing

SQL Server installation, or used to upgrade an earlier version of SSIS.

**What tool should be used to change or modify the SSIS service account?**

SQL Server Configuration Manager – but check, this may have changed

**SSISDB Catalog**

Specialized database dedicated to SSIS. The SSISDB catalog serves as the SSIS project and package repository and is the recommended deployment target for SQL Server 2012 SSIS solutions.

**SSISDB settings**

|  |  |
| --- | --- |
| Encryption Algorithm Name | The type of encryption algorithm that is used to encrypt sensitive data. |
| Clean Logs Periodically | When the value is True (default), operation details and operation messages older than the number of days specified by the Retention Period property are deleted from the catalog. When the value is False, all operation details and operation messages remain stored. |
| Retention Period (days) | The number of days (365 by default) that operation details and operation messages are stored in the catalog. |
| Server-wide Default Logging Level | The default logging level for the Integration Services server. |
| Maximum Number of Versions per Project | The number of new project versions that will be retained for a single project. |
| Periodically Remove Old Versions | When the value is True (default), only the number of project versions specified by the Maximum Number of Versions per Project property are stored in the catalog. |
| Validation Timeout | Validations will be stopped if they do not complete in the number of seconds specified by this property. |