

Intro to SQL Server Integration Services & SQL Agent Jobs

Stu Yarfitz
Analyst/Programmer
Collaborative Data Services
FHCRC
6/18/2008

SQL Server Integration Services (SSIS)

- A feature of SQL Server 2005
- Latest incarnation of Data Transformation Services (DTS)
- Used to transform and move data into and out of files and databases

Development Environment

- Visual Studio (Business Intelligence Development Studio)
- Visual Studio 2005, .NET Framework 2.0
- SQL Server Integration Services components need to be installed in both Visual Studio and SQL Server

Integration Services Project

- Developed within a Visual Studio Solution
- A project can contain one or more SSIS packages

SSIS Packages

- Basic unit of SSIS
- Replaces DTS packages from earlier versions of SQL Server.
- Migration tool for DTS packages
- .dtsx file type

Visual Studio SSIS Windows

- Standard
 - Solution Explorer
 - Properties
 - Toolbox
- SSIS specific
 - Major tabs for each package
 - Tabs within each package for views
 - Variables
 - Error List

Components of Packages

- Control flows
- Data flows
- Event handlers
- Variables, expressions, package configurations
- Connection managers

View - Control Flow

- Process steps or events
- Steps are linked together by precedence constraints
 - Value – success (green), failure (red), completion (blue)
 - Evaluation operation – constraint, expression
 - Multiple constraints - logical AND, OR

Control Flow Toolbox Items

- File system tasks (copy, delete, rename files, FTP)
- Execute tasks (SQL Stored Procedure, Windows process task , SSIS package task)
- Control structures (For loop container, foreach loop container)
- Data flow tasks

View - Data Flow

- Data flow processes are components of a data flow task
- Each data flow task in a control flow has its own set of data flow processes
- Data flow sources (flat file, OLE DB, XML, Excel)
- Data flow transformations (data conversion, derived column, merge, sort)
- Data flow destinations (SQL server, OLE DB, flat file, Excel)
- Each source and destination is associated with one or more connection managers

Connection Managers

- Control and data flow objects may require a connection manager
- Various types (OLE DB, flat file, ADO, Excel, OLE DB, FTP)
- Available properties depend on type
- All have a connection string
- Browse to create a connection string

Data Flow Sources & Destinations

- Data is strongly typed
- Editor – connection manager, columns, error handling
- Advanced editor – column mappings, data type properties, error handling
- Input and output column mappings are generated automatically
- Error handling can be defined for individual column and error type
- Errors can be directed to specific output files

Data Flow Transformations

- Derived Column Transformation Editor useful for adding new columns or changing data type
- Original and derived columns
- Expression builder

Variables

- User or system defined
- Scope can be package level or for a control flow step
- Specific data type
- Value can change at run time based on configuration setting

Expressions

- Many object properties can be controlled by expressions
- Expression builder - variables, functions, type casts, operators
- Can create complex equations incorporating variables, date functions, etc.

Package Configurations

- Allows for variable and property changes at runtime
- Configuration options
 - XML configuration file (.dtsConfig)
 - Environment variables
 - SQL Server table
- Properties can be set to read only

View - Event Handlers

- Event handler tasks can be defined for each executable
- Events include OnPostvalidate, OnTaskFailed, OnVariableValueChanged
- An error handler is itself a control flow and can include multiple steps

View - Package Explorer

- Provides access to all components
 - Variables
 - Precedence constraints
 - Connection managers
 - Executables

Event Logging

- In the SSIS menu
- Can define log files of diverse types (SQL Server, XML, text) and connection managers for them
- Can specify what processes are logged

Deployment

- Has a deployment utility and wizard
 - Project > properties
- Awkward to use and difficult to make changes in packages and configuration files after deployment
- My solution is to copy the package after final build to a subdirectory of SQL Server 2005

Package Execution Options

- From within Visual Studio
- SQL Server Management Studio
 - Integration Services
 - SQL Server Agent job
- Windows batch job using the dtexec utility

Integration Services Packages in SQL Server

- Integration services is a Windows service accessed via SQL Server Management Studio
- Stored packages
 - File system
 - MSDB – a SQL Server system database
 - SSIS Package Store
- It is convenient to define a package directory

Package Features in SQL Server

- Packages must be imported and re-imported if changed
 - Even though they are displayed when copied to the package directory
- Runtime Options
 - Packages lack configuration files – these can be added at run time
 - Command files can be specified
 - Connection manager connection strings
 - Other options

SQL Server Agent

- Runs as a Windows service
- Accessed through SQL Server Management Studio
- Runs SQL Server Agent jobs

SQL Server Agent Jobs

- A job has an owner
- A job needs to be enabled
- A job is comprised of steps
- Jobs can be scheduled

Job Steps

- Correspond to SSIS packages
- Can be other things (i.e. T-SQL script, SQL Server Analysis Services Query)
- Are executed in a defined sequence
 - Specified actions occur upon success or failure
- Package source (SSIS Package Store, SQL Server, File System)
- Configurations – paths to .dtsConfig files
 - Passwords for secure packages
- Other specifications – command files, logging, connection managers

Job Schedules

- Can have multiple schedules for a job
- Can run multiple jobs under one schedule

History Files

- Right-click Jobs or specific job to view
- Logs SQL Agents jobs – very useful

SQL Server Agent Security and Permissions

- Complex security model incorporating Windows, SQL Server, and SSIS components
- Most importantly, SQL Server Agent doesn't have permissions to run jobs!
- A user who is logged in and has the necessary permissions can usually run a job through Management Studio

Use Proxies ...

- The preferable way to run jobs is through proxies
- A proxy is based on a Windows user account that has adequate permissions for the files and directories involved
- The Windows user is assigned a SQL Server login with adequate permissions for the databases involved

...Proxies

- A SQL Server credential is created for the user (its password is saved)
- A SQL Server proxy is created based on the credential, and assigned to the necessary subsystems (SSIS, Operating System)
- Each step in a job must be run as a user (proxy) with the right permissions

Package Level Security ...

- Security is also enforced at the package level
- Package Security ProtectionLevel property
- Default is EncryptSensitiveWithUserKey
- The default won't work for running packages with sensitive information (passwords for example) as part of scheduled jobs!
 - The protection level needs to be set to EncryptSensitiveWithPassword

... Package Level Security

- This is particularly important for packages including SSH components or executing external SSH processes such as SFTP
- Package level security should not be reset in Visual Studio since it causes Visual Studio to crash

Setting Package Level Security

- Should be set at the time that a package is imported or re-imported through Management Studio
- In the Import Package dialog box the Protection level has a non-enabled text box followed by a button with a dot
- Click on the button, and change the package protection level to EncryptSensitiveWithPassword
- Enter and confirm the password for the package
- This password will need to be entered when the package is configured as a step in a SQL Agent job