



## Overview of SSIS

---

There are many articles on SSIS focusing on different aspects of SSIS.

When I started learning SSIS I came across many great articles and found a solution to almost all of the problems I faced. This article is nothing more than compiling all the blog/help files and providing a comprehensive view of SSIS.

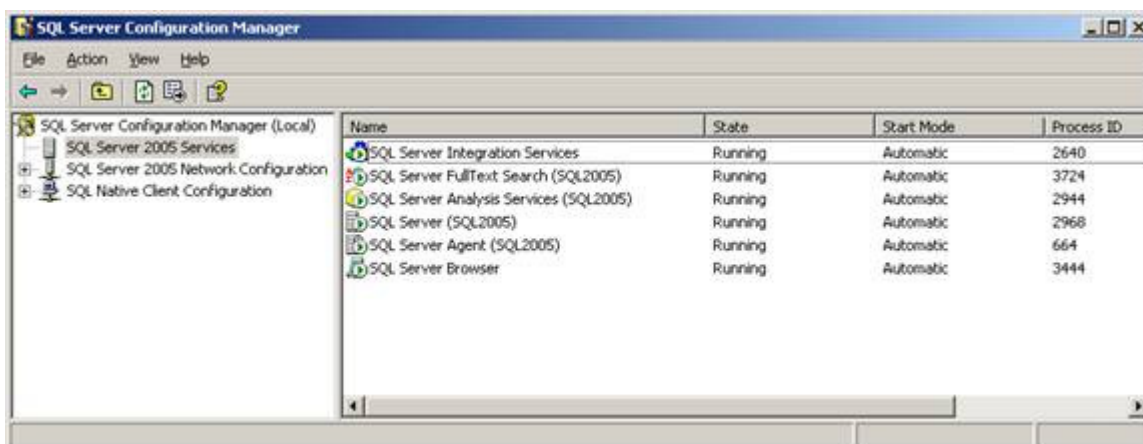
### Let's start

SSIS is not an upgrade to DTS. It is a replacement for DTS. SSIS should not be considered just a part of SQL Server. It is very powerful and should be treated with respect.

SSIS Service is installed as a **single service** on the server even if we have multiple **instances** of SQL Server. In other words it is a shared service across multiple instances.

We can manage the SSIS service via SQL Server Configuration Manager

Start -> Program -> SQL Server 2005 -> Configuration Tools -> SQL Server Configuration Manager



### SSIS Tools we can use

#### SQL Server Management Studio (SMS)

- Migrates DTS packages into SQL Server 2005
- Run/ Schedule Packages
- Assign Package Security
- View Packages

**BIDS ("SQL Server Business Intelligence Development Studio")** - This is nothing but Visual Studio.

Imagine the number of hours spent in a meeting in a conference room at Microsoft to come up with this name.

- Manage, Develop and Edit Package
- Deploy Package

## SQL Server Configuration Manager

- To manage SSIS service

## Accessing SSIS via SQL Server Management Studio (SSMS)

We can connect to SSIS Service for various reasons via SSMS. To connect to SSIS we have to mention the server name without the instance of SQL Server. SSMS will connect to the default instance of SQL Server (if specified, else it would error out)

Now the question is how can we manage packages on different Instances?

For that we need to change a configuration file. By default, the file is located in the folder, Program Files\Microsoft SQL Server\90\DTS\Binn, and the file name is MsDtsSrvr.ini.xml.

The default configuration file contains the following settings:

- For Integration Services in SMS (Object Explorer) are the MSDB and File System folders.
- The packages in the file system that the SSIS Service manages are located in Program Files\Microsoft SQL Server\90\DTS\Packages.

You can modify the configuration file to display additional folders in Object Explorer, or to specify a different folder or additional folders in the file system to be managed by SSIS service. The example below shows how to configure the MsDtsSrvr.ini.xml to use more than one MSDB database, which are stored in different database instances.

Here is configuration file before and after the changes. Before:

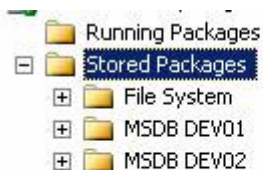
```
<?xml version="1.0" encoding="utf-8"?>
<DtsServiceConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<StopExecutingPackagesOnShutdown>true</StopExecutingPackagesOnShutdown>
  <TopLevelFolders>
    <Folder xsi:type="SqlServerFolder">
      <Name>MSDB</Name>
      <ServerName>.</ServerName>
    </Folder>
    <Folder xsi:type="FileSystemFolder">
      <Name>File System</Name>
      <StorePath>..\Packages</StorePath>
    </Folder>
  </TopLevelFolders>
</DtsServiceConfiguration>
```

After:

```
<?xml version="1.0"encoding="utf-8"?>
<DtsServiceConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <StopExecutingPackagesOnShutdown>true</StopExecutingPackagesOnShutdown>
  <TopLevelFolders>
    <Folder xsi:type="SqlServerFolder">
      <Name>MSDB DEV01</Name>
      <Server Name>\<Instance Name 1></ServerName>
    </Folder>
    <Folder xsi:type="SqlServerFolder">
      <Name>MSDB DEV02</Name>
      <Server Name>\<Instance Name 2></ServerName>
    </Folder>
    <Folder xsi:type="FileSystemFolder">
      <Name>File System</Name>
      <StorePath>..\Packages</StorePath>
    </Folder>
  </TopLevelFolders>
</DtsServiceConfiguration>
```

**Note: We have to restart the Integration Service after the changes are done to the file**

You will now notice that we can see 2 different folders for MSDB



With this, we can manage multiple instances of SQL Server for SSIS. Check this link :

<http://bloggingabout.net/blogs/mglaser/archive/2007/03/01/multiple-sql-server-integration-services-ssis-database-instances-on-one-machine.aspx>

This can also give us the advantage to manage the Packages for different instance from a different location.

To access SSIS remotely we should have RPC port open (Port 135). Here are few links which can help in case you are not able to connect remotely:

[http://sqljunkies.com/WebLog/knight\\_reign/archive/2006/01/05/17769.aspx](http://sqljunkies.com/WebLog/knight_reign/archive/2006/01/05/17769.aspx) and  
[http://www.sqlteam.com/forums/topic.asp?TOPIC\\_ID=62541](http://www.sqlteam.com/forums/topic.asp?TOPIC_ID=62541)

## Executing the SSIS package from Command Prompt

The **dtexec** command prompt utility is used to configure and execute SQL Server 2005 Integration Services (SSIS) packages. The **dtexec** utility provides access to all the package configuration and execution features, such as connections, properties, variables, logging, and progress indicators. The **dtexec** utility lets you load packages from three sources: a Microsoft SQL Server database, the SSIS service, and the file system.

The command of dtexec is very simple, just a matter of getting used to it. Most commonly used will be

the /set command which is used to assign values to the properties of the package / task or to assign values to variables.

Need more on dtexec utility? Try this link: <http://msdn2.microsoft.com/en-us/library/ms162810.aspx>

To get a better idea of the variables and properties and how to assign them from the command line we will go through a small example.

Let's add a new data source. Right Click Data Source -> New Data Source -> Complete the Wizard (Use SQL server Authentication)

Go to the properties window for the newly created data source

Note: To view properties window use "F4"

In the expression properties of the data source add ConnectionString as property and the following line in the expression.

```
"Data Source=" + @[User::ServerName] + ";User ID=" + @[User::UserName]
+ ";Password=" + @[User::Password] + ";Initial Catalog=msdb;Provider=SQLOLEDB.1;"
```

You will notice we have 3 variables in the expression

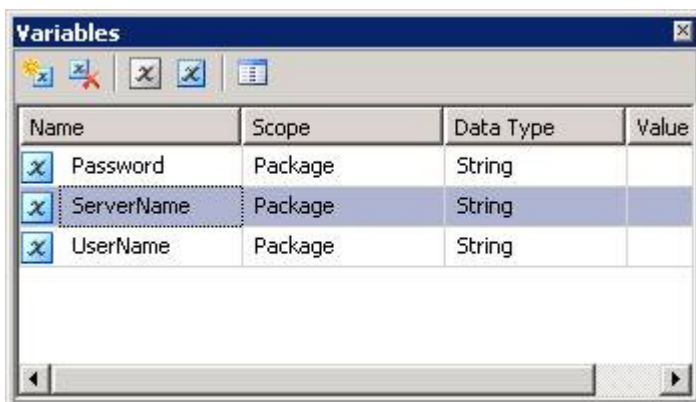
User::ServerName

User::UserName

User::Password

We will add the above variable and then assign the values from command line. To add variables open the variable window (2 ways to open variables window)

- SSIS -> Variables
- View -> Other Window -> Variables



Now in the variables window add the above variables, just keep the scope as default ("Package") and change the data type to "String". Now we can change the value of the variable from the command prompt, which will change the connectionstring of our data source.

## Syntax for command line

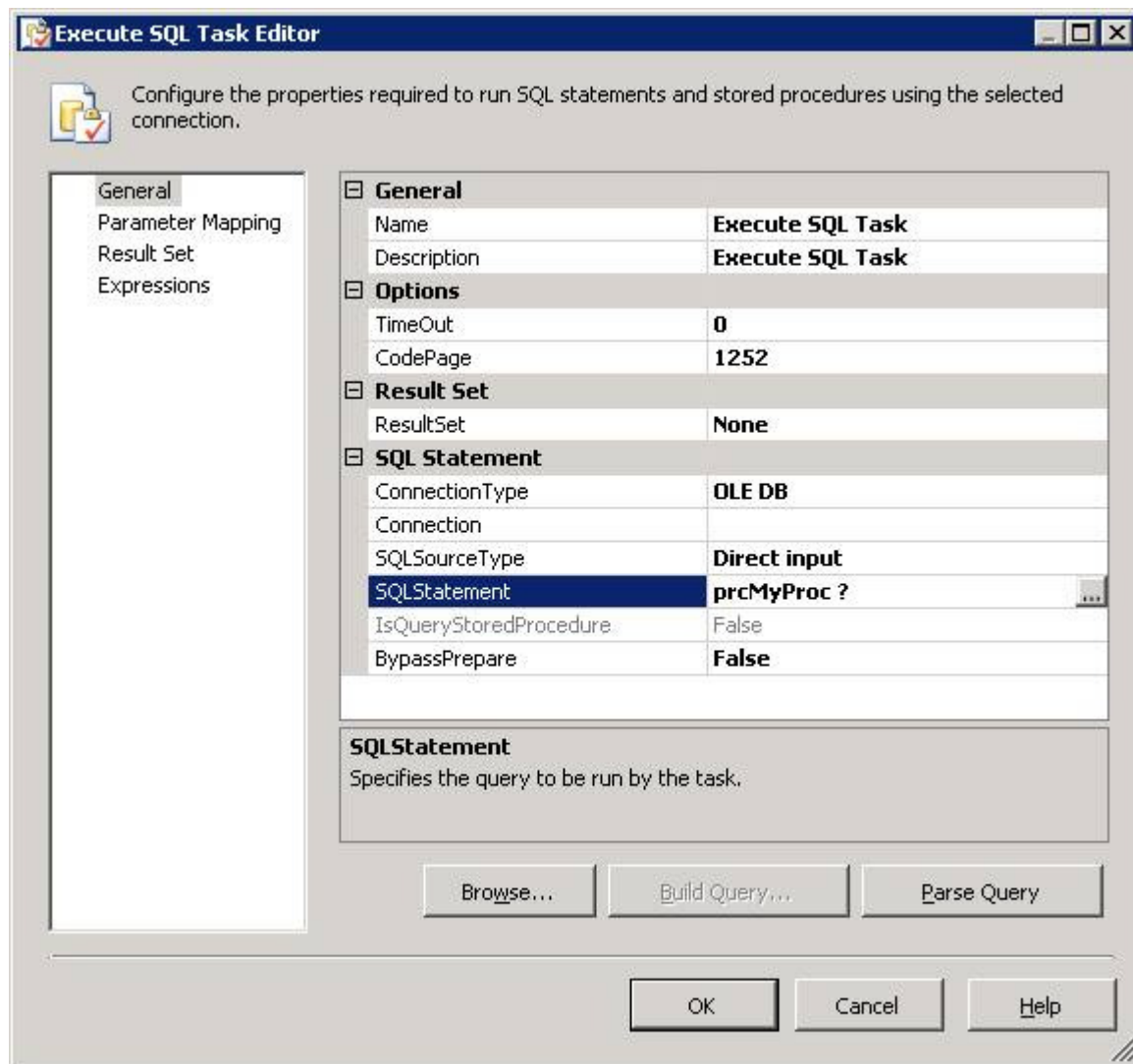
```
dtexec /f "D:\PacakageName"
/set \Package.Variables[User::ServerName].Properties[Value];< ServerName>
/set \Package.Variables[User::UserName].Properties[Value];<UserName>
/set \Package.Variables[User::Password].Properties[Value];< Password>
```

Other method of changing the connectionstring without the variables or expression is to manipulate the property of our data source directly. Here is the command line syntax:

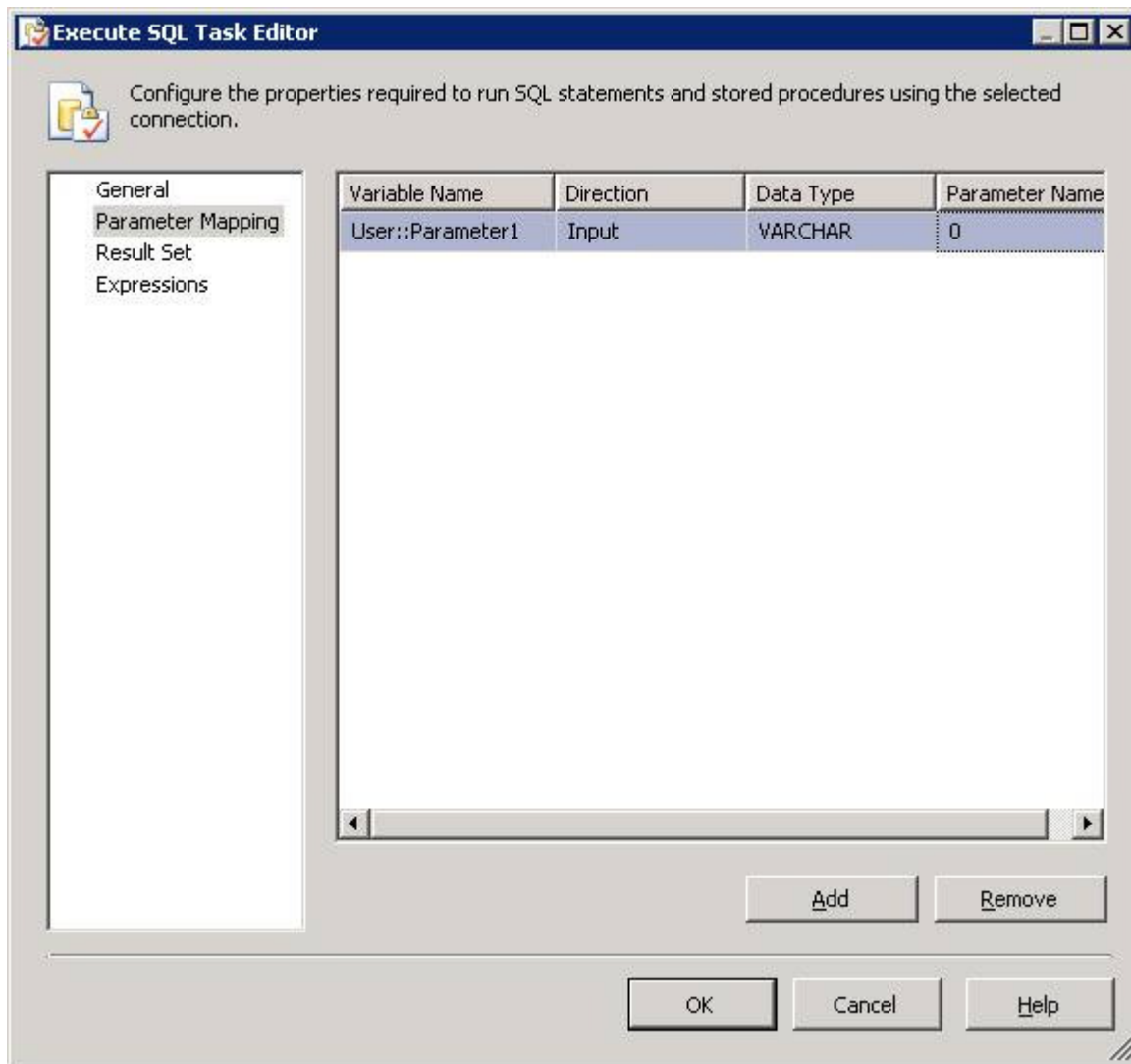
```
dtexec /f " D:\PacakageName"
/set \Package.Connections[DataSourceName].Properties[ServerName];<ServerName>
/set \Package.Connections[DataSourceName].Properties[UserName];<UserName>
/set \Package.Connections[DataSourceName].Properties[Password];<Password>
```

### h3 class="Section"Passing variable as a parameter in a stored procedure

Add a new SQL Server Task and go to the property window (F4 key)



Add the proc name in SQLStatement and add "?" for the parameter and then go to parameter Mapping (make sure you have the connection set in the previous screen)

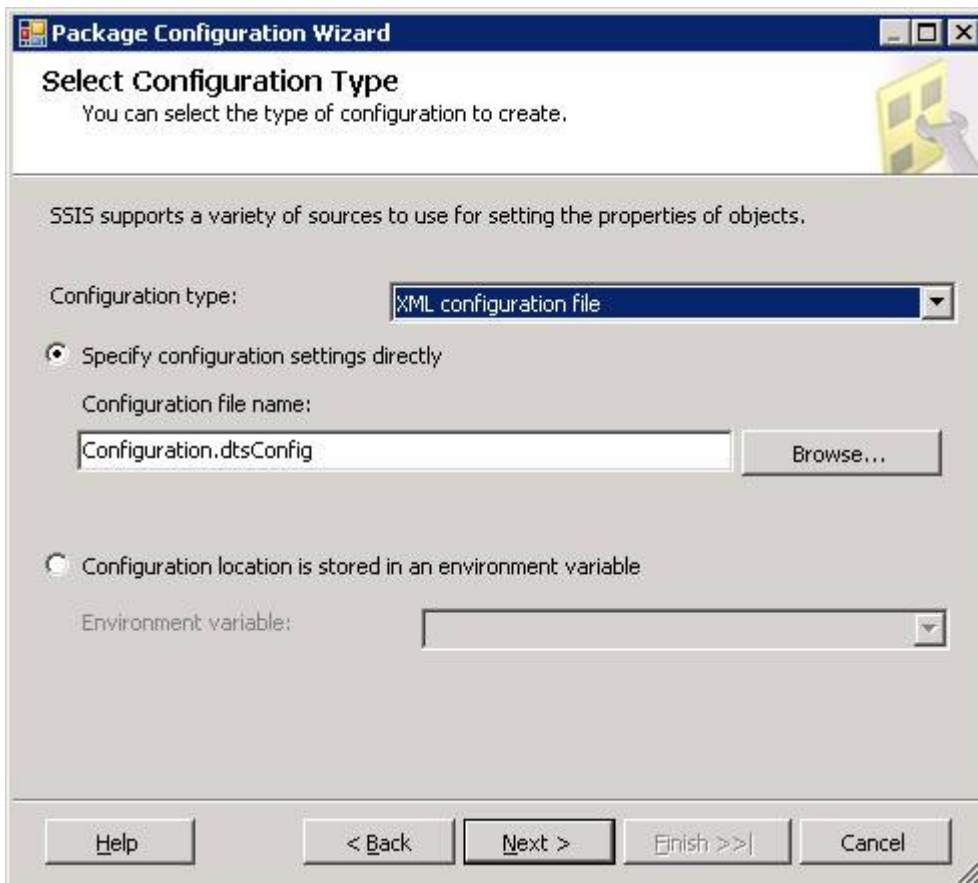


Add a new parameter Mapping, change to the appropriate Data Type and the Parameter Name to 0 (this determine the ordinal position).

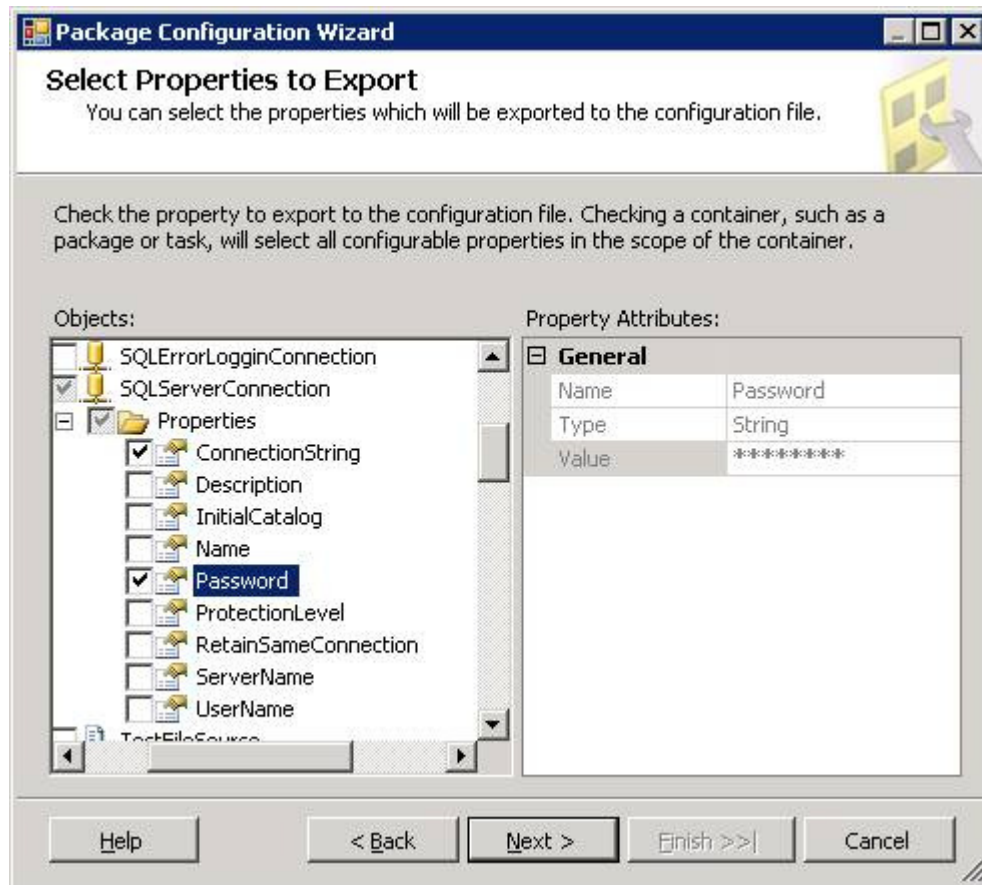
For more information on Parameter Mapping: <http://msdn2.microsoft.com/en-us/library/ms187685.aspx>

The best way to find the property path of a property is to create an XML configuration file.

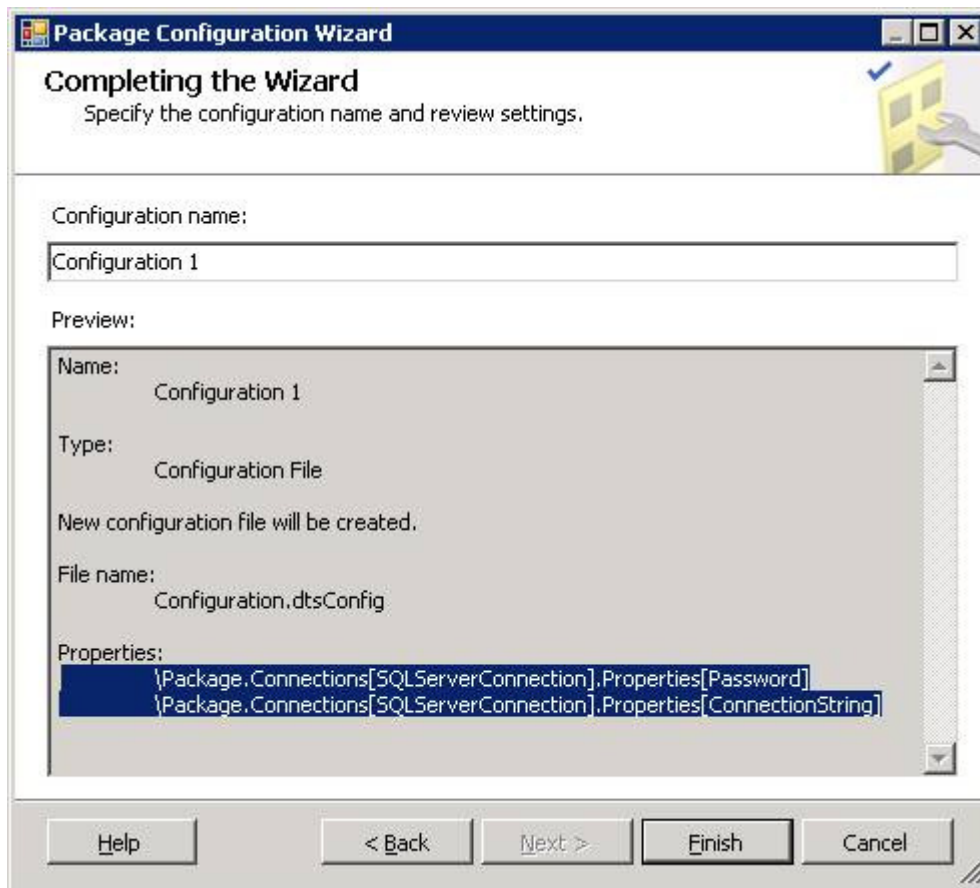
SSIS -> Package Configuration



Select the property in question







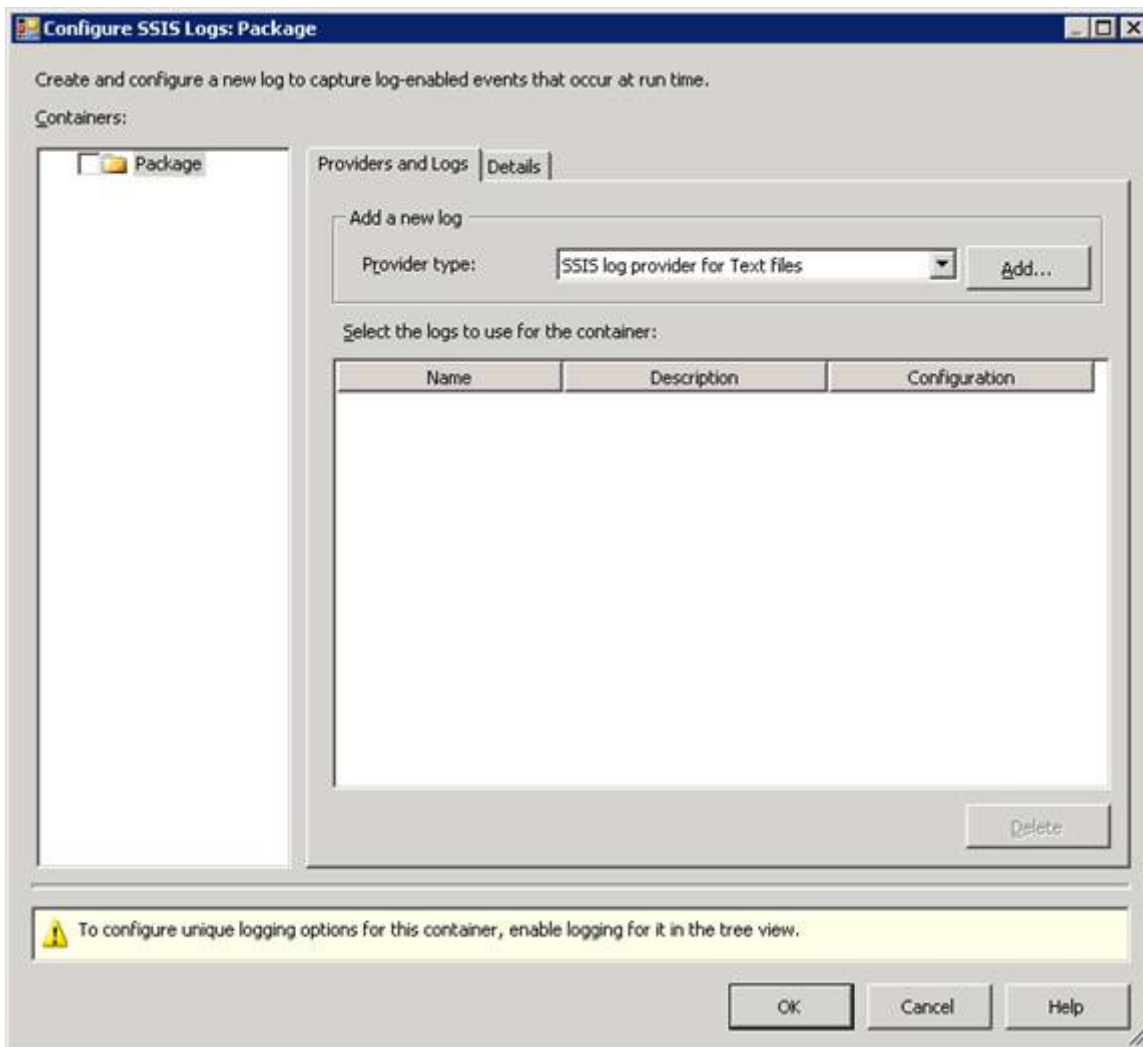
## Logging Package Information

SSIS comes with some predefined ways to log package information. We can log the information to the following 5 providers (excluding the log window)

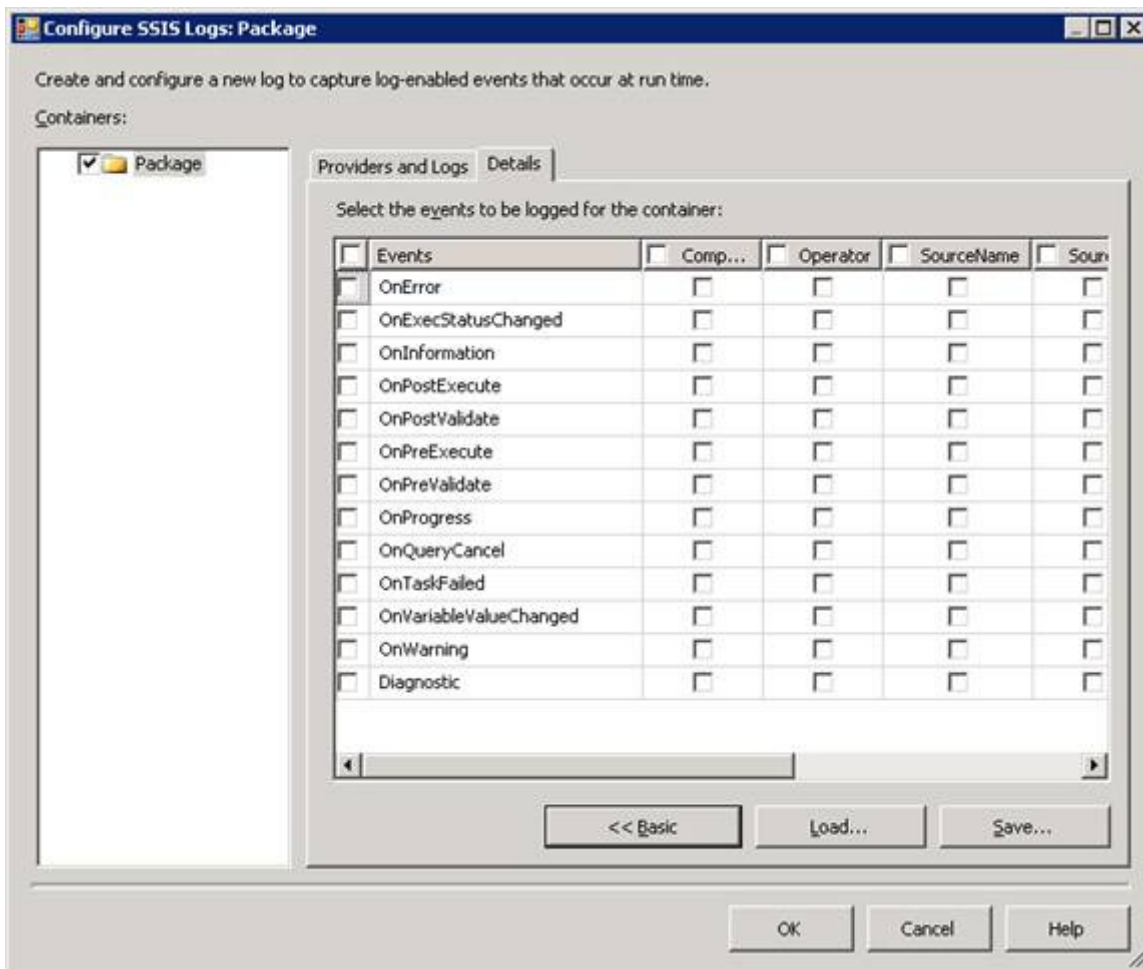
- Text File
- SQL Server Profiler
- SQL Server (Table)
- Windows Event Log
- XML File.

Logging is disabled by default. We have to enable it at the package level. We can log the information at the package level or at the task level.

To enable logging go to the property window of package and change the LoggingMode to enable. You can change the loggingMode at the task level based on your requirement. Now to select the destination for logging information we need to right click on the control flow (anywhere) and then click Logging, which will open the following window



Choose the provider type and then Click Add. Complete the wizards for that provider and the logging will happen. If you click on the details tab you will notice here we can select the events and the information which we need to capture



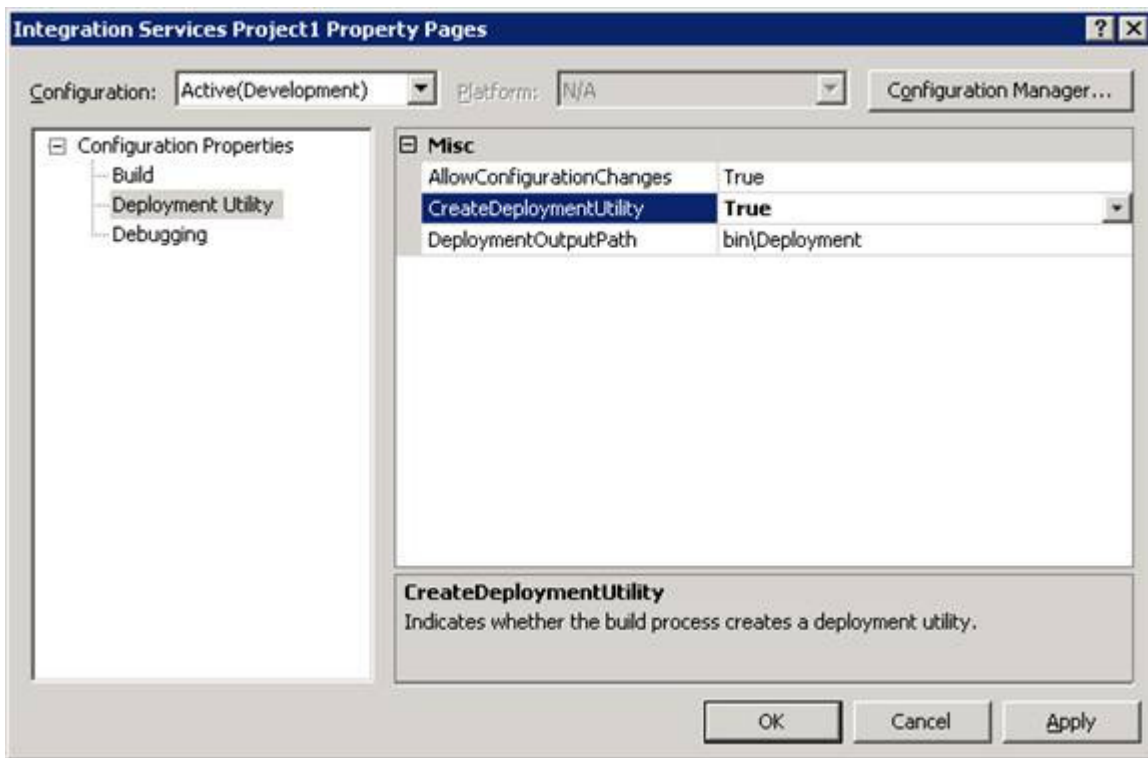
We can save this configuration in XML by clicking the save or can apply the configuration based on previously saved XML file by clicking Load.

One thing confusing, was the name of the table which logs the information? The name of the table is sysdtslog90 (it will be created in the database which was specified in the connection string). If the table is already created it will append the information.

## Deploying the package

Once the package is ready for production we can deploy using the following steps.

Go to the package Property Projects -> Project Properties which will open the following window



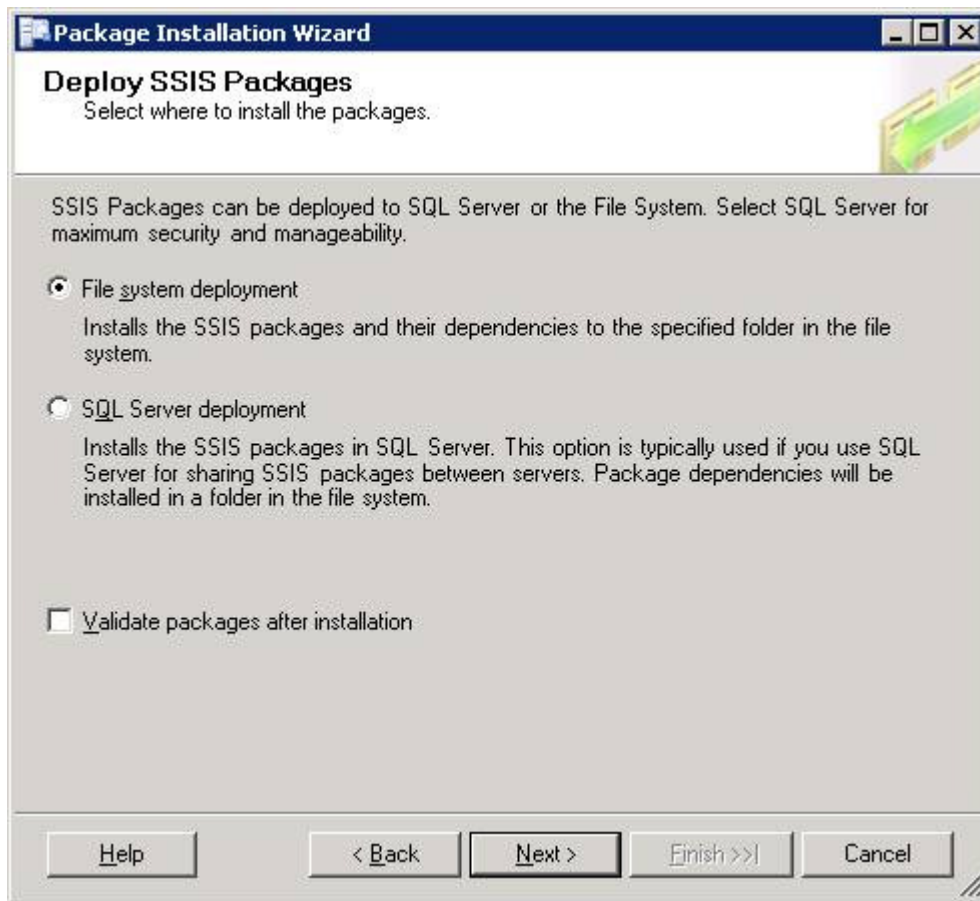
By default the CreateDeploymentUtility will be false, change it to True and hit ok. Now Build the Project Build -> Build <Project Name>

Now go to the location where you have the package (not sure where it is?) Check the property window of the package (press F4) and the location Path. Go to that folder from window explorer and you will see a folder as Bin and under that you will see the Deployment folder (this is the same path which you see in DeploymentOutputPath in the above screen)

You will see a file name (your project name); file type will be Development Manifest. Double click on the file and it will open the Deployment wizard



The next screen will give you 2 options, select the one were you want to deploy and complete the wizard.



## Running the package from a SQL Agent job

While adding a new step in the job now we have a new type as SSIS package

**New Job Step**

Select a page: **General** | Advanced

Script | Help

Step name: \_\_\_\_\_

Type: **SQL Server Integration Services Package**

Run as: **SQL Agent Service Account**

Set values | Verification | Command line

General | Configurations | Command files | Data sources | Execution options | Logging

Package source: **SQL Server**

Server: \_\_\_\_\_

Log on to the server

☒ Use Windows Authentication

☐ Use SQL Server Authentication

User name: \_\_\_\_\_

Password: \_\_\_\_\_

Package: \_\_\_\_\_

Connection

Server: K12-DBSTG1\DEV02

Connection: dts\_sa

[View connection properties](#)

Progress

Ready

OK Cancel

Complete the above screen and we are set to go.

### Simple trick

Ever tried using the enter key in an annotation. Try and see it will not work. Now try Ctrl + Enter.

Copyright © 2002-2007 Red Gate Network. All Rights Reserved.