**Package Configuration Steps**

In BIDS, create a new SSIS project called PackageConfigExample.

**2.** In the default Package.dtsx package, create a new data connection to the SQL Server AdventureWorks database.

**3.** Go to properties of Control Flow >Configurations>Collections

**4.** In the Package Configurations Organizer dialog box, select Enable Package Configurations.

**5.** Click the Add button to add a new configuration.

**6.** Click Next on the Package Configuration Wizard welcome screen.

**7.** On the Select Configuration Type screen, for the configuration type, select XML Configuration File

Keep “Specify configuration settings directly” selected.

This means the configuration will expect the XML file to be in the same file location wherever the package is loaded. Use the other option, “Configuration location is stored in an environment

variable,” if you expect this location to change.

For the configuration filename, type

foldername\test.dtsConfig. A new file will be created if one does not already exist. Click

Next to continue.

**8.** On the Select Properties to Export screen, under the \Package\Connections folder, expand

your SQL Server instance connection. Expand the Properties folder, check the Connection-

String, InitialCatalog, and ServerName properties,

Then click Next

**9.** On the Completing the Wizard screen, type **SQL Server Connection Change** in the Configuration

Name text box. Click Finish.

**10.** Your new configuration will now appear in the Package Configuration Organizer. Click Close.

**11.** Open the configuration file from the location you specified. Here, you can change the three

property values. the ConfiguredValue tags outline the values

for each configurable property. If you change the property in the XML file and then load the

package to a new project, the SSIS package will inherit the properties based on the XML file.

<?xml version="1.0"?>

<DTSConfiguration>

<DTSConfigurationHeading>

<DTSConfigurationFileInfo

GeneratedBy="JOEPROD\Owner" GeneratedFromPackageName="Package"

GeneratedFromPackageID=

"{FA099B09-C230-4688-AF93-E88C73C8683F}" GeneratedDate="1/1/2005 9:28:29 PM"/>

</DTSConfigurationHeading>

<Configuration ConfiguredType="Property"

Path="\Package.Connections[{C2FB43E4-2DD7-44D0-B616-

D9EF80D60901}].ConnectionString" ValueType="String">

**<ConfiguredValue>Data Source=localhost;Initial Catalog=AdventureWorks;**

**Provider=SQLNCLI10.1;Integrated Security=SSPI;**

**Auto Translate=False;</ConfiguredValue>**

</Configuration>

<Configuration ConfiguredType="Property" Path="\Package.Connections

[{C2FB43E4-2DD7-44D0-B616-D9EF80D60901}].InitialCatalog" ValueType="String">

<ConfiguredValue>**AdventureWorks**</ConfiguredValue>

</Configuration>

<Configuration ConfiguredType="Property" Path=

"\Package.Connections[{C2FB43E4-2DD7-44D0-B616-D9EF80D60901}].ServerName"

ValueType="String">

<ConfiguredValue>**localhost**</ConfiguredValue>

</Configuration>

</DTSConfiguration>

Instead of XML select SQL Server option

Specify connection and click on configuration table

Specify a name for configuration filter

Click on Next

Check the properties to export to configuration table

Click on Next

Specify the Configuration Name

Click on Finish

Goto SQL Server

Goto the database which you have configured

You should see dbo.SSIS configuration table

Select \* from SSIS configuration table

You will see the properties which you have selected earlier

Only the **XML Configuration File** and **SQL Server** configuration types support including multiple properties in a configuration.

**Static Options**

**Configuration type**

Select the type of source in which to store the configuration, using the following options:

|  |  |
| --- | --- |
| **Value** | **Description** |
| **XML configuration file** | Store the configuration as an XML file. Selecting this value displays the dynamic options in the section, **Configuration Type**. |
| **Environment variable** | Store the configuration in one of the environment variables. Selecting this value displays the dynamic options in the section, **Configuration Type**. |
| **Registry entry** | Store the configuration in the registry. Selecting this value displays the dynamic options in the section, **Configuration Type**. |
| **Parent package variable** | Store the configuration as a variable in the package that contains the task. Selecting this value displays the dynamic options in the section, **Configuration Type**. |
| **SQL Server** | Store the configuration in a table in SQL Server. Selecting this value displays the dynamic options in the section, **Configuration Type**. |

**Next**

View the next page in the wizard sequence.

**ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10is_5techref/local/collapse.gifDynamic Options**

**Configuration Type Option = XML Configuration File**

**Specify configuration settings directly**

Use to specify settings directly.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Configuration file name** | Type the path of the configuration file that the wizard generates. |
| **Browse** | Use the **Select Configuration File Location** dialog box to specify the path of the configuration file that the wizard generates. If the file does not exist, it is created by the wizard. |

**Configuration location is stored in an environment variable**

Use to specify the environment variable in which to store the configuration.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Environment variable** | Select an environment variable from the list. |

**Configuration Type Option = Environment Variable**

**Environment variable**

Select the environment variable that contains the configuration information.

**Configuration Type Option = Registry Entry**

**Specify configuration settings directly**

Use to specify settings directly.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Registry entry** | Type the registry key that contains the configuration information. The format is <registry key>.  The registry key must already exist in HKEY\_CURRENT\_USER and have a value named Value. The value can be a DWORD or a string.  If you want to use a registry key is not at the root of HKEY\_CURRENT\_USER, use the format <registry key\registry key\...> to identify the key. |

**Configuration location is stored in an environment variable**

Use to specify the environment variable to store the configuration in.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Environment variable** | Select an environment variable from the list. |

**Configuration Type Option = Parent Package Variable**

**Specify configuration settings directly**

Use to specify settings directly.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Parent variable** | Specify the variable in the parent package that contains the configuration information. |

**Configuration location is stored in an environment variable**

Use to specify the environment variable that stores the configuration.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Environment variable** | Select an environment variable from the list. |

**Configuration Type Options = SQL Server**

**Specify configuration settings directly**

Use to specify settings directly.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Connection** | Select a connection from the list, or click **New** to create a new connection. |
| **Configuration table** | Select an existing table, or click **New** to write a SQL statement that creates a new table. |
| **Configuration filter** | Select an existing configuration name or type a new name.  Many SQL Server configurations can be stored in the same table, and each configuration can include multiple configuration items.  This user-defined value is stored in the table to identify configuration items that belong to a particular configuration |

**Configuration location is stored in an environment variable**

Use to specify the environment variable where the configuration is stored.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Environment variable** | Select an environment variable from the list. |