**Execute a SQL Server SSIS package**

In SQL Server 2005 and higher versions there are different ways in which one can execute an SSIS package. Let us go through each option one by one.

**Using .NET**

using Microsoft.SqlServer.Dts.Runtime;

private void Execute\_Package()

{

string pkgLocation = @"c:\test.dtsx";

Application app = new Application();

Package pkg= app.LoadPackage(pkgLocation, null);

DTSExecResult pkgResults;

Variables vars = pkg.Variables;

vars["A\_Variable"].Value = "Some value";

DTSExecResult pkgResults = pkg.Execute(null, vars, null, null, null);

pkg.Execute();

if (pkgResults == DTSExecResult.Success)

Console.WriteLine("Package ran successfully");

else

Console.WriteLine("Package failed");

}

using System;

using System.Collections.Generic;

using System.Text;

using Microsoft.SqlServer.Dts.Runtime;

namespace ExecuteSSIS

{

class Program

{

static void Main(string[] args)

{

Application app = new Application();

*//*

*//* ***Load package from file system***

*//*

Package package = app.LoadPackage("c:\\ExamplePackage.dtsx", null);

package.ImportConfigurationFile("c:\\ExamplePackage.dtsConfig");

Variables vars = package.Variables;

vars["MyVariable"].Value = "value from c#";

DTSExecResult result = package.Execute();

Console.WriteLine("Package Execution results: {0}",result.ToString());

*//*

*//* ***Load package from SQL Server***

*//*

Package package2 = app.LoadFromSqlServer(

"ExamplePackage","server\_name", "sa", "your\_password", null);

package2.ImportConfigurationFile("c:\\ExamplePackage.dtsConfig");

Variables vars2 = package2.Variables;

vars2["MyVariable"].Value = "value from c# again";

DTSExecResult result2 = package2.Execute();

Console.WriteLine("Package Execution results: {0}", result2.ToString());

}

}

}

using Microsoft.SqlServer.Dts.Runtime;

protected void btnExecute\_Click(object sender, EventArgs e)

{

Application app = new Application();

Package package = null;

try

{

string fileName =

Server.MapPath(System.IO.Path.GetFileName(FileUpload1.PostedFile.FileName.ToString()));

FileUpload1.PostedFile.SaveAs(fileName);

*//Load DTSX*

package =

app.LoadPackage(@&quot;D:\SSIS\_ASP\_NET\SSIS\_ASP\_NET\_DEMO\SSIS\_ASP\_NET\_DEMO\Package1.dtsx&quot;, null);

*//Global Package Variable*

Variables vars = package.Variables;

vars[&quot;Business\_ID&quot;].Value = txtBusinessID.Text;

vars[&quot;Business\_Name&quot;].Value = txtBusinessName.Text;

*//Specify Excel Connection From DTSX Connection Manager*

package.Connections[&quot;SourceConnectionExcel&quot;].ConnectionString =

&quot;provider=Microsoft.Jet.OLEDB.4.0;data source=&quot; + fileName + &quot;;Extended Properties=Excel 8.0; &quot;;

*//Execute DTSX.*

Microsoft.SqlServer.Dts.Runtime.DTSExecResult results = package.Execute();

}

catch (Exception ex)

{

throw ex;

}

finally

{

package.Dispose();

package = null;

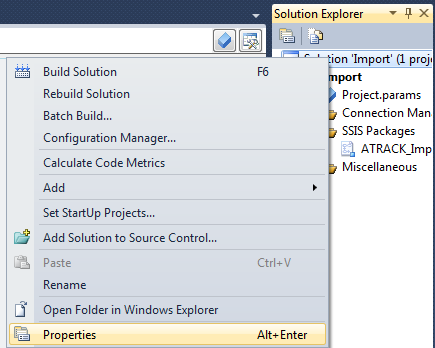
}

}

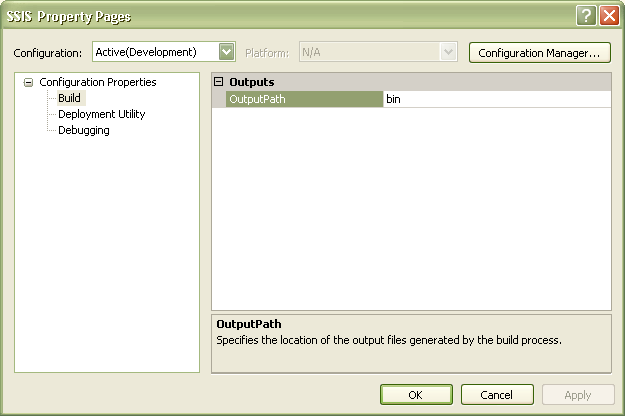
**Execute SSIS Package Using SQL Server Business Intelligence Development Studio (BIDS)**

During the development phase of the project developers can test the SSIS package execution by running the package from Business Intelligence Development Studio a.k.a. BIDS.

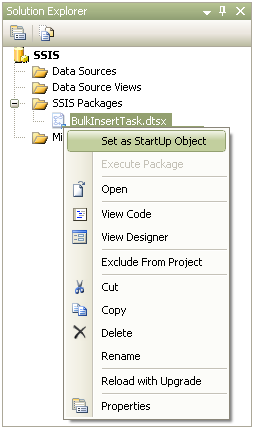
1. In Solution Explorer, right click the SSIS project folder that contains the package which you want to run and then click properties as shown in the snippet below.



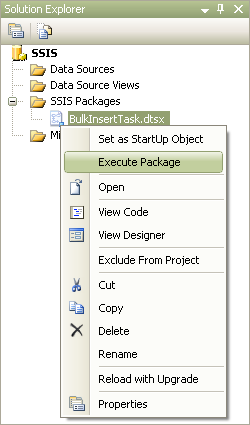
2. In the SSIS Property Pages dialog box, select Build option under the Configuration Properties node and in the right side panel, provide the folder location where you want the SSIS package to be deployed within the OutputPath. Click OK to save the changes in the property page.



3. In Solution Explorer, right click the SSIS Package and then click Set as Startup Object option as shown in the snippet below.



4. Finally to execute the SSIS package, right click the package within Solution Explorer and select Execute Package option from the drop down menu as shown in the snippet below.



**Execute SSIS Package using DTEXEC.EXE Command Line Utility**

The syntax to execute a SSIS package which is stored in a File System is shown below.

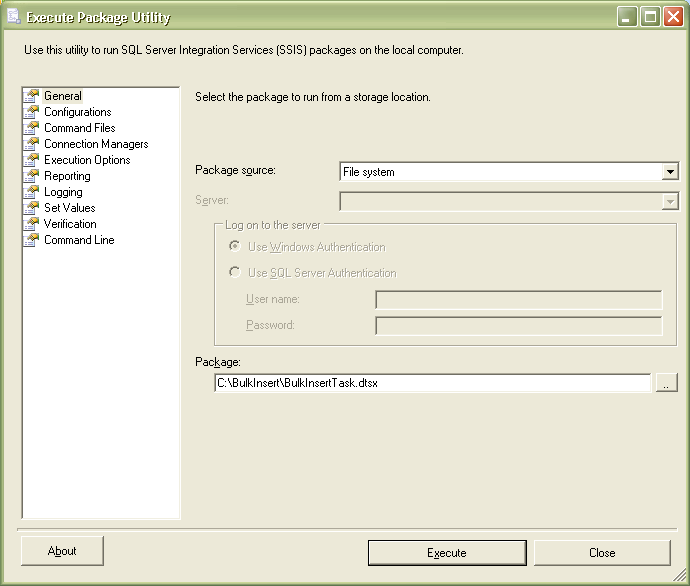
**DTEXEC.EXE /F "C:\BulkInsert\BulkInsertTask.dtsx"**



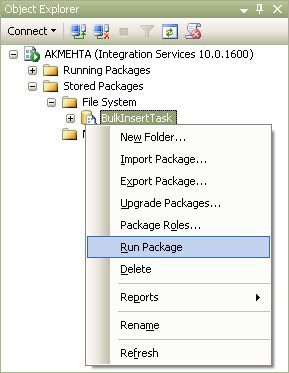
**Execute SSIS Package using DTEXECUI.EXE Utility**

Using the Execute Package Utility (DTEXECUI.EXE) graphical interface one can execute an SSIS package that is stored in a File System, SQL Server or an SSIS Package Store.

1. In command line, type **DTEXECUI.EXE** which will open up Execute Package Utility as shown in the snippet below. Within the Execute Package Utility, click on the General tab and then choose the Package source as “**File System**”, next you need to provide the path of the SSIS package under Package option and finally click the Execute button to execute the SSIS package.



The Execute Package Utility is also used when you execute the SSIS package from the Integration Services node in SQL Server Management Studio.

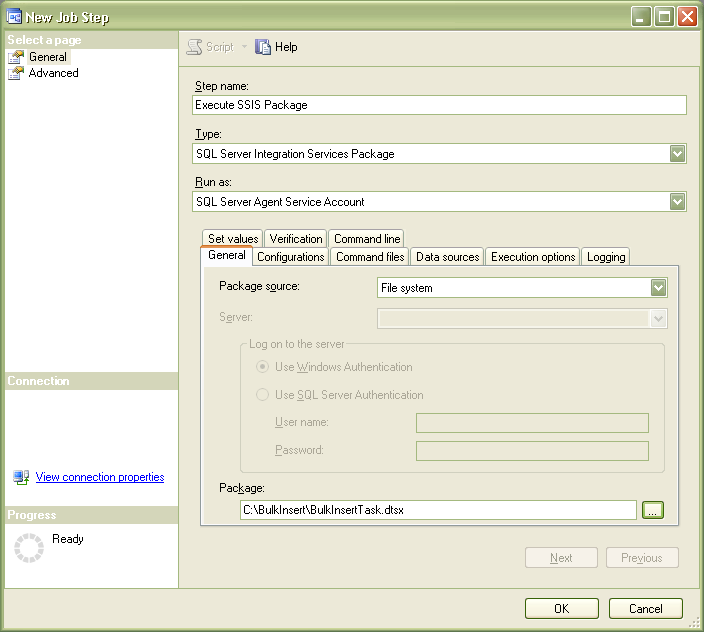


**Execute SSIS Package using SQL Server Agent Job**

Using a SQL Server Agent Job one can execute an SSIS package that is stored in a File System, SQL Server or an SSIS Package Store. This can be done by creating a new SQL Server Agent Job and then by adding a new step with details as mentioned in the snippet below.

1. In New Job Step dialog box provide an appropriate Step name, then choose “**SQL Server Integration Services Package**” option as Type from the drop down list, and then choose “**SQL Server Agent Service Account**” as Run as value.

2. In the General tab choose the File System as Package Source and provide the location of the SSIS package under Package option.



3. Click OK to save the job step and click OK once again to save the SQL Server Agent Job