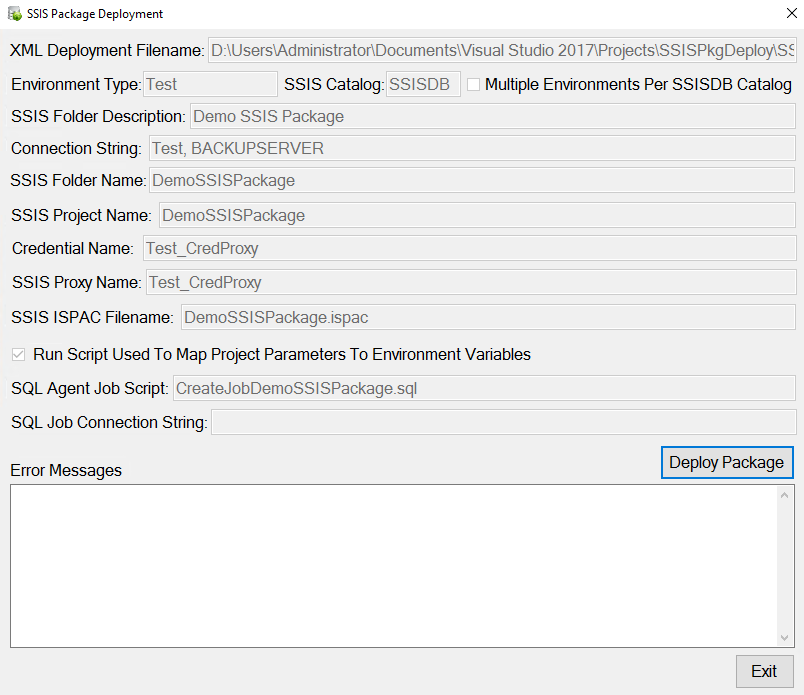
**SSIS Package Deployment & SQL Job Installation**

Facilitates SSIS Package installation, configuring Package settings, and SQL Agent Job deployment using a C# Windows Form utility program SSISPkgDeploy.exe. Most of the settings for the deployment are located in an XML file.

Prerequisites for running this program are having the .Net Framework 4.5.2 installed, and the utility uses integrated security to connect to the target deployment server(s). So, the user running the program needs to have access to the Integration Services Catalogs (SSISDB), master database to configure the SSIS packages, and msdb database to install the SQL Agent Jobs. Also, the SSISDB must exist on the target server, the utility will not create the SSISDB.

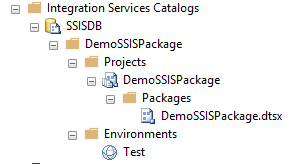
The deployment utility is shown in the below screen shot. It opens showing the deployment settings from the application configuration file (SSISPkgDeploy.exe.config). The deployment is run by clicking the “Deploy Package” button, and the status of the deployment is shown in the “Error Messages” textbox. The Error Messages textbox will also show any system errors that occur during the deployment. If there is an error during deployment, the text in the Error Messages textbox will turn red. The “Deploy Package” button is disable once clicked, so the deployment can only be run once, and the “Exit” button turns green if the deployment runs successfully.



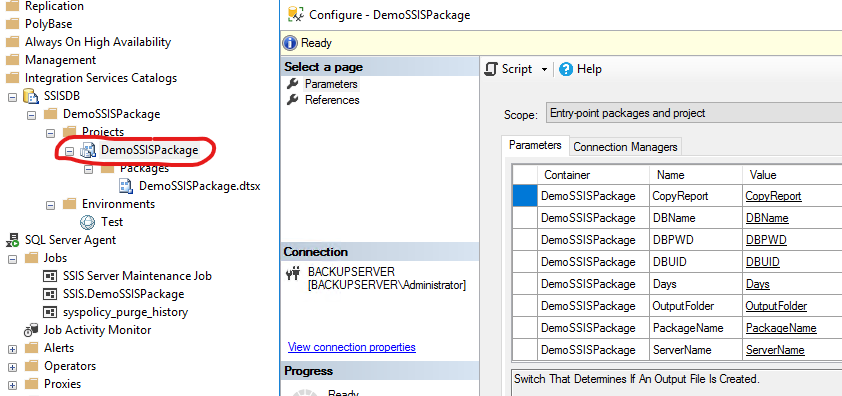
The main features of the deployment utility are described below:

* **Environment Type**: Can setup SSIS Project Environments for Test, Dev, QA, UATPri, UATSec, PRDPri and PRDSec. Allows deployment to a single environment, or a pipe delimited string of environments can be specified, so you can deploy to multiple environments. For example, you can deploy to the PRDPri and PRDSec environments, which are the Production Primary and Secondary servers when an Availability Group is setup. This allows consistent deployments to multiple environments.
* **SSIS Catalog**: Assumes SSISDB, this is the Microsoft recommended practice.
* **Multiple Environments Per SSISDB Catalog:** When checked (true), packages will be deployed to folders by environment, so different code can be run in each environment. Useful when deploying to a single server where you want packages and agent jobs running against multiple environments (e.g Dev, QA, UAT). Setting this option to “true”, will append the environment name to the SSIS Folder Name, and also allow creating separate agent jobs for each environment.
* **SSIS Folder Description**: Description of the folder created in the Integration Services Catalog for the Project Deployment.
* **Connection String**: Connection string to the master database on the server where the SSIS packages are to be deployed. The master database is used for SSISDB Folder and Project changes. Initially shows a list of the Environment Names and deployment database servers, but shows the environment connection string when the deployment is running. The connection string assumes integrated security.
* **SSIS Folder Name**: Name of the folder to create in SSISDB for the deployment. Can differ from the Visual Studio project name. The environment name (e.g. Dev, QA, UAT, PRD) is appended to this name when “Multiple Environments Per SSISDB Catalog” is checked.
* **SSIS Project Name**: Name of the project in the SSISDB where the SSIS packages are deployed. Needs to be the same as the Visual Studio project name.
* **SSIS Credential Name**: Name of the Credential used to create the Proxy a SQL Agent Job will run under.
* **SSIS Proxy Name**: Name of the proxy account used when running the SQL Agent Job(s). If specified in the XML file, a Credential and Proxy Account can be created during deployment.
* **SSIS ISPAC Filename**: Name of the ISPAC file created by the Visual Studio SSDT project which contains the SSIS packages to deploy. The ISPAC file needs to be in the same folder as the utility file SSISPkgDeploy.exe.
* **Run Script Used To Map Project Parameters To Environment Variables**: Run a SQL script used to map project variables used by the SSIS packages to Project Environment variables created by the utility.
* **SQL Agent Job Script:** SQL script file used to create the SQL Agent Jobs associated with the SSIS packages deployed. The job script is run with a fixed list of parameters so that the job can be configured to run for a specific proxy and environment. The script file needs to be in the same folder as the utility file SSISPkgDeploy.exe.
* **SQL Job Connection String:** Connection string to the msdb database on the server where the SSIS packages are to be deployed. Field is initially blank, then shows the environment connection string when the deployment is running. The msdb database is used for SQL Agent Jobs.

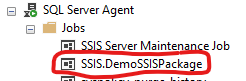
The deployment example shown above will install the Demo SSIS package to the Test Environment on server BACKUPSERVER using variables as shown below:



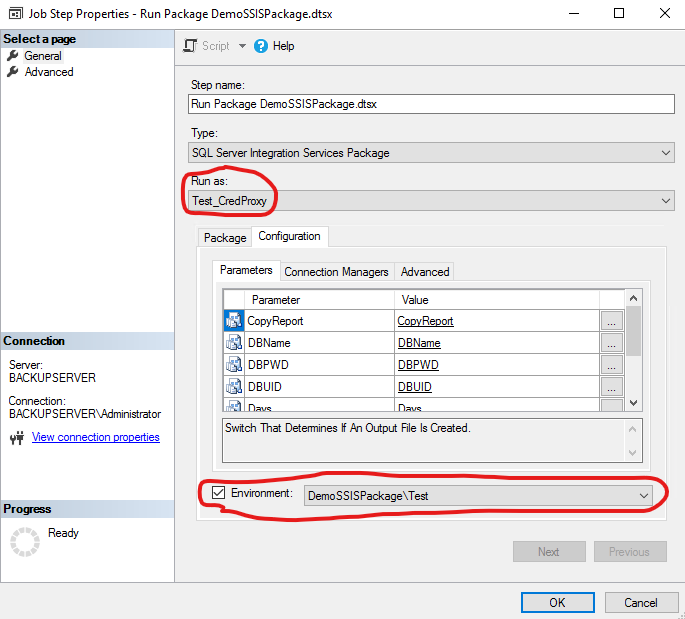
The deployment will also map SSIS package project parameters, to the Test Environment variables using a SQL script, as shown below:



The utility runs script file CreateJobDemoSSISPackage.sql to create the below SQL Agent Job:



The SQL Agent Jobs will also be configured to use the Test Environment variables as shown below. Also, the SQL Job was setup to Run using a Proxy Account (Test\_CredProxy).



**Deployment XML File DemoSSISPackage.xml**

Deployment specific configuration settings are setup using an XML file (DemoSSISPackage.xml) as shown below. This is the where all the settings used for deployment maintained. The configuration needs to be setup for each specific deployment. Deployment only works for the “Project Deployment Model” supported in SQL Server 2012 and greater. Using the XML configuration shown below allows enough flexibility to accommodate most deployment scenarios. You can deploy just the SSIS packages to a single environment. Or, deploy SSIS packages, create and populate Environments for Project Parameters, map an Environment to a Project, and run a script to create SQL Agent Job(s), setup the jobs to run for a specific Proxy and Environment variables, for multiple environments located on different servers, with a single button click!

<?xml version="1.0" encoding="utf-8"?>

<deployment>

<projectSettings>

<!-- EnvironmentType: Test, Dev, QA, UATPri, UATSec, PRDPri, PRDSec.-->

<!-- Can deploy to multiple environments per run using pipe delimited string like: value="UATPri|UATSec" -->

<setting name="EnvironmentType" value="Test" />

<setting name="SSISCatalog" value="SSISDB" />

<setting name="MultiEnvPerCatalog" value="false" />

<setting name="SSISFolderName" value="DemoSSISPackage" />

<setting name="SSISFolderDescription" value="Demo SSIS Package" />

<setting name="SSISProjectName" value="DemoSSISPackage" />

<!-- If use="true" the ISPAC file will be deployed. -->

<setting name="SSISProjectFilename" value="DemoSSISPackage.ispac" use="true" />

<setting name="MapProjParamsToEnvVar" value="true" />

<!-- If use="true" the SQL Agent Job Script(s) are executed. -->

<setting name="SQLAgentJobScript" value="CreateJobDemoSSISPackage.sql" use="true" />

</projectSettings>

<proxyCredentialSetup>

<proxy name="CredentialName" value="Test\_CredProxy" />

<!-- If use="true" the proxy name can be used in the SQL Agent Job Script(s). -->

<proxy name="SSISProxyName" value="Test\_CredProxy" use="true" />

<proxy name="CreateCredential" value="true" />

<proxy name="CreateProxy" value="true" />

<proxy name="SameIdentitySecretForAllEnv" value="true" />

</proxyCredentialSetup>

<environmentServerNames>

<environment name="Test" server="BACKUPSERVER" />

<environment name="Dev" server="BACKUPSERVER" />

<environment name="QA" server="BACKUPSERVER" />

<environment name="UATPri" server="BACKUPSERVER" />

<environment name="UATSec" server="BACKUPSERVER" />

<environment name="PRDPri" server="BACKUPSERVER" />

<environment name="PRDSec" server="BACKUPSERVER" />

</environmentServerNames>

<environmentVariablesTest>

<!-- Variable Types: Boolean, Byte, Char, DateTime, DBNull, Decimal, Double, Empty, Int16, Int32, Int64, Object, SByte, Single, String, UInt16, UInt32, UInt64 -->

<variable name="DBName" type="String" value="SSISDB" sensitive="false" description="Connection String Database Name" />

<variable name="DBPWD" type="String" value="" sensitive="true" description="Connection String Password" />

<variable name="DBUID" type="String" value="" sensitive="true" description="Connection String User ID, Leave Blank For Integrated Security" />

<variable name="ServerName" type="String" value="BACKUPSERVER" sensitive="false" description="Connection String Database Server Name" />

<variable name="CopyReport" type="Boolean" value="true" sensitive="false" description="Switch That Determines If An Output File Is Created." />

<variable name="Days" type="Int16" value="-1" sensitive="false" description="Number Of Days To Report On. Needs To Be A Negative Number." />

<variable name="OutputFolder" type="String" value="D:\" sensitive="false" description="Folder Where The Output File Is Copied. Needs To Be A Local Folder, Or Share Accessable By &quot;Everyone&quot;." />

<variable name="PackageName" type="String" value="DemoSSISPackage.ispac" sensitive="false" description="Name Of The SSIS Package To Report On." />

</environmentVariablesTest>

<environmentVariablesDev>

<!-- Variable Types: Boolean, Byte, Char, DateTime, DBNull, Decimal, Double, Empty, Int16, Int32, Int64, Object, SByte, Single, String, UInt16, UInt32, UInt64 -->

</environmentVariablesDev>

<environmentVariablesQA>

<!-- Variable Types: Boolean, Byte, Char, DateTime, DBNull, Decimal, Double, Empty, Int16, Int32, Int64, Object, SByte, Single, String, UInt16, UInt32, UInt64 -->

<variable name="DBName" type="String" value="SSISDB" sensitive="false" description="Connection String Database Name" />

<variable name="DBPWD" type="String" value="" sensitive="true" description="Connection String Password" />

<variable name="DBUID" type="String" value="" sensitive="true" description="Connection String User ID, Leave Blank For Integrated Security" />

<variable name="ServerName" type="String" value="BACKUPSERVER" sensitive="false" description="Connection String Database Server Name" />

<variable name="CopyReport" type="Boolean" value="true" sensitive="false" description="Switch That Determines If An Output File Is Created." />

<variable name="Days" type="Int16" value="-1" sensitive="false" description="Number Of Days To Report On. Needs To Be A Negative Number." />

<variable name="OutputFolder" type="String" value="D:\" sensitive="false" description="Folder Where The Output File Is Copied. Needs To Be A Local Folder, Or Share Accessable By &quot;Everyone&quot;." />

<variable name="PackageName" type="String" value="DemoSSISPackage.dtsx" sensitive="false" description="Name Of The SSIS Package To Report On." />

</environmentVariablesQA>

<environmentVariablesUATPri>

<!-- Variable Types: Boolean, Byte, Char, DateTime, DBNull, Decimal, Double, Empty, Int16, Int32, Int64, Object, SByte, Single, String, UInt16, UInt32, UInt64 -->

</environmentVariablesUATPri>

<environmentVariablesUATSec>

<!-- Variable Types: Boolean, Byte, Char, DateTime, DBNull, Decimal, Double, Empty, Int16, Int32, Int64, Object, SByte, Single, String, UInt16, UInt32, UInt64 -->

</environmentVariablesUATSec>

<environmentVariablesPRDPri>

<!-- Variable Types: Boolean, Byte, Char, DateTime, DBNull, Decimal, Double, Empty, Int16, Int32, Int64, Object, SByte, Single, String, UInt16, UInt32, UInt64 -->

<variable name="DBName" type="String" value="SSISDB" sensitive="false" description="Connection String Database Name" />

<variable name="DBPWD" type="String" value="" sensitive="true" description="Connection String Password" />

<variable name="DBUID" type="String" value="" sensitive="true" description="Connection String User ID, Leave Blank For Integrated Security" />

<variable name="ServerName" type="String" value="BACKUPSERVER" sensitive="false" description="Connection String Database Server Name" />

<variable name="CopyReport" type="Boolean" value="false" sensitive="false" description="Switch That Determines If An Output File Is Created." />

<variable name="Days" type="Int16" value="-1" sensitive="false" description="Number Of Days To Report On. Needs To Be A Negative Number." />

<variable name="OutputFolder" type="String" value="D:\" sensitive="false" description="Folder Where The Output File Is Copied. Needs To Be A Local Folder, Or Share Accessable By &quot;Everyone&quot;." />

<variable name="PackageName" type="String" value="DemoSSISPackage.dtsx" sensitive="false" description="Name Of The SSIS Package To Report On." />

</environmentVariablesPRDPri>

<environmentVariablesPRDSec>

<!-- Variable Types: Boolean, Byte, Char, DateTime, DBNull, Decimal, Double, Empty, Int16, Int32, Int64, Object, SByte, Single, String, UInt16, UInt32, UInt64 -->

</environmentVariablesPRDSec>

</deployment>

**Project App Conf File SSISPkgDeploy.exe.config**

There are a few settings in the app config file that can be used to control deployment. The LogFileName key controls the name of the application log file. The log file is saved in the application folder. The XSDValidationFile key is used for the XSD file used to validate the deployment XML file. The XSDValidationFile needs to be in the application folder. The DeployXMLFile key is an optional setting. If the DeployXMLFile exists in the InitialDirectory folder, then the application will start using this XML file for deployment parameters. Otherwise, a file picker is used to select the deployment XML file. The InitialDirectory key is also optional, if the InitialDirectory exists, the application will default to looking for deployment XML files in the InitialDirectory.

<appSettings>

<add key="LogFileName" value="SSISLogFile.txt" />

<add key="XSDValidationFile" value="SSISPkgDeploy.xsd" />

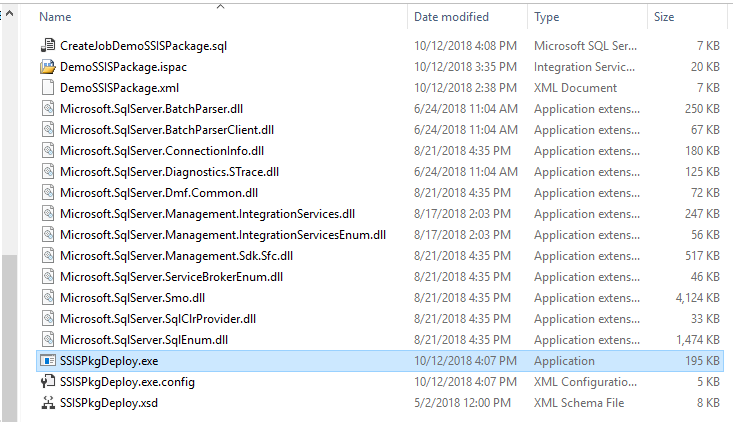
<add key="DeployXMLFile" value="DemoSSISPackage.xml" />

<add key="InitialDirectory" value="D:\Users\Administrator\Documents\Visual Studio 2017\Projects\SSISPkgDeploy\SSISPkgDeploy\bin\Debug\" />

</appSettings>

**Practical Implementation Example**

An example of how the tool is implemented for a specific deployment is shown below. All the files for a deployment would be copied to a folder (e.g. D:\Users\Administrator\Documents\Visual Studio 2017\Projects\SSISPkgDeploy\SSISPkgDeploy\bin\Debug\). The utility is a Visual Studio C# Windows Form application. So, you need to deploy the EXE (SSISPkgDeploy.exe), the app config file (SSISPkgDeploy.exe.config), associated DLL’s that are referenced by the project (e.g. Microsoft.SqlServer.BatchParser.dll, Microsoft.SqlServer.Management.IntegrationServices.dll), the ISPAC file for the SSIS Project Deployment (DemoSSISPackage.ispac), and the SQL Agent Job script (CreateJobDemoSSISPackage.sql).



**Logging**

The utility logs run information to a configurable log file. The app config file has a setting to name the log file (<add key="LogFileName" value="SSISLogFile.txt" />). Information that is sent to the “Error Messages” textbox in the utility is written to the log file at the end of the utility run. The below shows an example of a log entry from a deployment run.

====================================================================================================

Time Logged: Thursday, October 18, 2018 10:27:42 PM

Deployment for Environment Type: 'Test', Server Name: BACKUPSERVER.

Multiple Environments Per SSISDB Catalog: False.

Deploying D:\Users\Administrator\Documents\Visual Studio 2017\Projects\SSISPkgDeploy\SSISPkgDeploy\bin\Debug\DemoSSISPackage.ispac project ISPAC file.

SSIS Project (DemoSSISPackage.ispac) has been successfully deployed!

SSIS 'Test' Environment has been successfully created!

Added Environment Variable: DBName, Type = String, Value = SSISDB, Description = Connection String Database Name, Sensitive = false

Added Environment Variable: DBPWD, Type = String, Value = , Description = Connection String Password, Sensitive = true

Added Environment Variable: DBUID, Type = String, Value = , Description = Connection String User ID, Leave Blank For Integrated Security, Sensitive = true

Added Environment Variable: ServerName, Type = String, Value = BACKUPSERVER, Description = Connection String Database Server Name, Sensitive = false

Added Environment Variable: CopyReport, Type = Boolean, Value = true, Description = Switch That Determines If An Output File Is Created., Sensitive = false

Added Environment Variable: Days, Type = Int16, Value = -1, Description = Number Of Days To Report On. Needs To Be A Negative Number., Sensitive = false

Added Environment Variable: OutputFolder, Type = String, Value = D:\, Description = Folder Where The Output File Is Copied. Needs To Be A Local Folder, Or Share Accessable By "Everyone"., Sensitive = false

Added Environment Variable: PackageName, Type = String, Value = DemoSSISPackage.dtsx, Description = Name Of The SSIS Package To Report On., Sensitive = false

Environment reference 'Test' has been added to the SSIS Project DemoSSISPackage

Script used to create the SQL job CREDENTIAL was run.

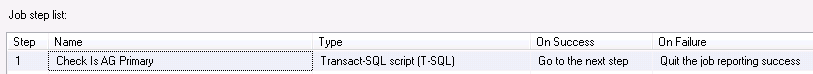
Script used to create the SQL job PROXY was run.

Script used to map Project Parameters to Environment Variables was run.

SQL Agent Job Script File D:\Users\Administrator\Documents\Visual Studio 2017\Projects\SSISPkgDeploy\SSISPkgDeploy\bin\Debug\CreateJobDemoSSISPackage.sql was run. ====================================================================================================

**Dealing With Availability Groups**

The utility can deploy packages to both Primary and Secondary servers when Availability Groups are setup. This applies to UAT and Production environments. The Environment Types UATPri & UATSec (UAT Primary and Secondary, and PRDPri & PRDSec (Production Primary and Secondary) can be used to deploy packages to the Primary and Secondary servers in an Availability Group. Theoretically, it is possible to just install to the listener. However, when packages are installed to servers participating in an Availability Group, you would typically only want the package running on the Primary server. To make sure the package is only running on the Primary, an extra Job step is needed that determines if the server is Primary, if so, go to the next Job step. Otherwise, quit the job reporting success. See example job step and script below:



-- FYI: This step is needed for Jobs running on servers setup with Availability Groups (AG).

-- Packages and Jobs are deployed to both the Primary and Secondary AG servers.

-- Allows Jobs to only run on the Primary server.

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobstep

@job\_id=@bJobID,

@step\_name=N'Check Is AG Primary',

@step\_id = 1,

@cmdexec\_success\_code = 0,

@on\_success\_action = 3,

@on\_success\_step\_id = 0,

@on\_fail\_action = 1,

@on\_fail\_step\_id = 0,

@retry\_attempts = 0,

@retry\_interval = 0,

@os\_run\_priority = 0,

@subsystem = N'TSQL',

@command =

N'-- Returns 1 if the database on the current instance is the primary replica. Otherwise returns 0.

IF (ISNULL([master].sys.fn\_hadr\_is\_primary\_replica(N''SSISDB''), 1) <> 1) -- NULL when Availability Groups not setup (e.g. Dev QA), continue to next step.

RAISERROR(''Not the PRIMARY server for this job, exiting with SUCCESS'',11, 1);',

@database\_name = N'master',

@flags = 0;