

[http://www.sqlservercentral.com/articles/Integration+Services+\(SSIS\)/125585/](http://www.sqlservercentral.com/articles/Integration+Services+(SSIS)/125585/)

Printed 2017/01/16 05:33AM

How to run SSIS Project as a SQL Job

By [Philip Horan](#), 2015/04/28

Introduction

In Part 1 of this series, [How to FTP a Dynamically Named Flat File](#), I explained how to create an SSIS Package that output a dynamically named flat file and then sent the output file to a FTP server. In this part, I will explain how to create an 'Integration Services Catalog' to deploy the SSIS Project created in the first article. I will then show how to run the SSIS Project on a schedule as SQL Server Job.

Scenario

I created an SSIS Project on my local machine, tested it and now wanted to run the package as a SQL Job. This needs to be on a schedule as I have done with other SSIS projects in the past. I fired up SQL Server Management Studio (SSMS) and connected to 'Integration Services'. I expanded the 'Stored Packages' folder, right clicked the 'MSDB' folder and selected 'Import Package'. I then proceeded to follow steps taken previously to import the package and create a SQL Job to run the package on a schedule.

I then sat back and gave myself a pat on the back for another job well done. My celebrations were a little premature as when I proceeded with the formality of executing the job it failed, reporting an error relating to being unable to process the FTP Task.

Message

Executed as user: DOMAIN\Account. Microsoft (R) SQL Server Execute Package Utility Version 12.0.2000.8 for 64-bit Copyright (C) Microsoft Corporation. All rights reserved. Started: 15:27:55 Error: 2015-03-26 15:27:56.31 Code: 0xC001602A Source: Package Connection manager "FTP Connection Manager" Description: An error occurred in the requested FTP operation. Detailed error description: The password was not allowed End Error Error: 2015-03-26 15:27:56.31 Code: 0xC002918F Source: FTP Task FTP Task Description: Unable to connect to FTP server using "FTP Connection Manager". End Error DTEXec: The package execution returned DTSER_FAILURE (1). Started: 15:27:55 Finished: 15:27:56 Elapsed: 0.375 seconds. The package execution failed. The step failed.

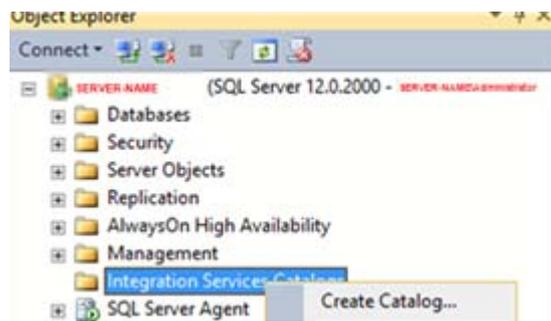
This got me thinking. I decided to seek some assistance on the [SQLServerCentral.com forums](#) and received some good advice from Phil Parkin. Phil also pointed me towards an article written by Andy Leonard, [SSIS Catalog Environments– Step 20 of the Stairway to Integration Services](#). The proceeding steps will act as a guide as to how I configured my SSIS Project to run as a SQL Job within SSMS and eliminate the error pertaining to the FTP Task above.

Note: For detailed information regards the options selected in the subsequent steps I thoroughly recommend that you read [SSIS Catalog Environments– Step 20 of the Stairway to Integration Services](#) by Andy Leonard.

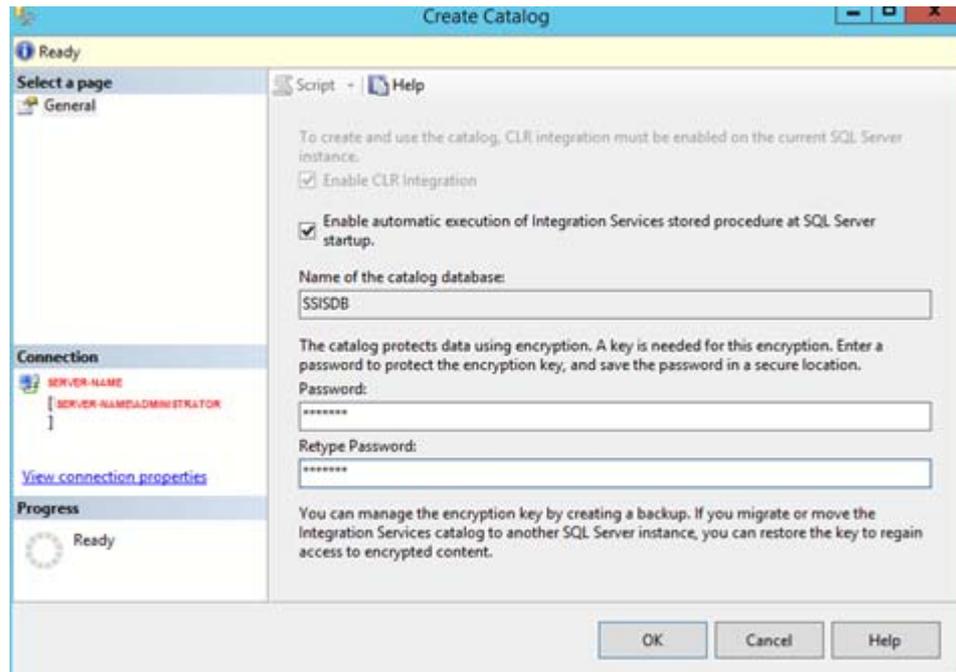
Step 1

In step 1 you will create an 'Integration Services Catalog'. The catalog will be used to 'Deploy' a SSIS

Project. Open SSMS then right click 'Integration Services Catalog'. Select 'Create Catalog'.



The 'Create Catalog' window opens. Select 'Enable CLR Integration' and 'Enable automatic execution...'. Provide a password of your choice when requested by the wizard.



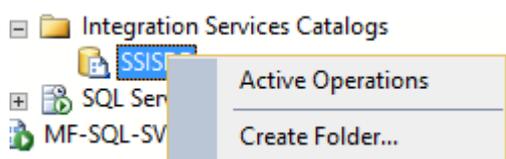
Click 'OK'. Upon completion the new SSIDB catalog will be listed as shown below.



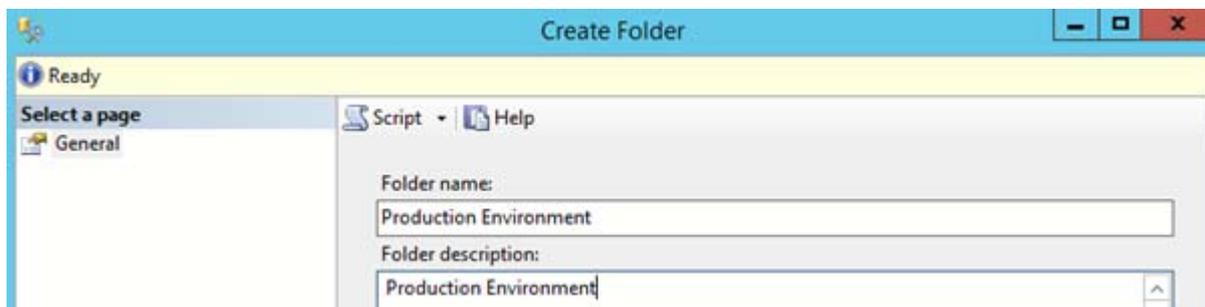
Step 2

In step 2 you will create a folder. This folder will store your 'SSIS Projects' and 'Environment Variables.' You can create multiple folders for your production, dev and test environments.

Right click and select 'Create Folder.'



You will see the Create Folder dialog. Enter "Production Environment" as shown below.

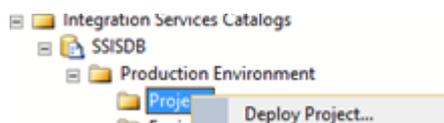


Two additional folders are created.

1. **Projects** - This is where we will deploy our SSIS project
2. **Environments** - This is where we will configure our variables to use to connect to FTP server

Step 3

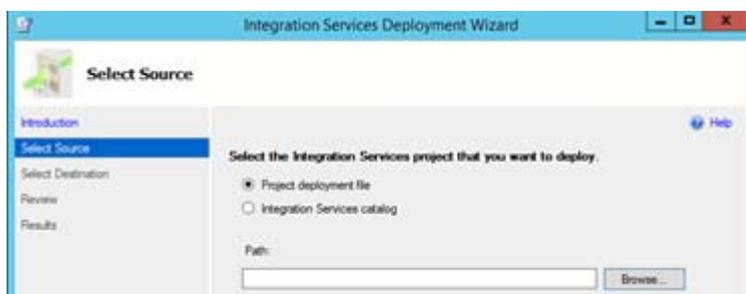
In this step you will 'Deploy' a 'SSIS Project' into the newly created SSISDB. For this you will need a file with a .ispac file extension. You can either use an existing project or select the .ispac file created if you followed Part 1 'How to FTP Dynamically Named Flat File'. Open SSMS. Right click the 'Projects' folder and select 'Deploy Project'.



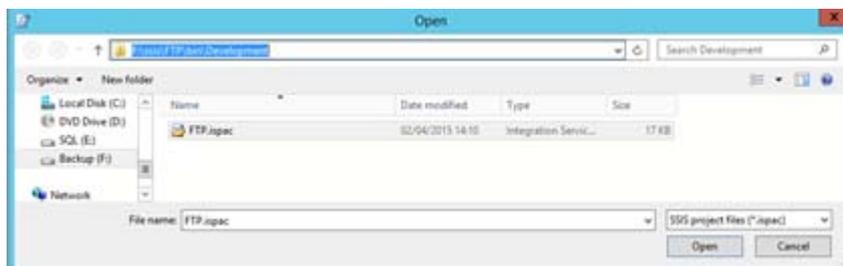
On selecting 'Deploy Project' the 'Integration Services Deployment Wizard' is launched.



Follow the wizard. When asked to select the integration project you want to deploy browse to your project deployment folder and select your .ispac file.



With the 'File name' field populated with your .ispac file select 'Open'.



Complete the steps presented by the wizard.

Select the Integration Services project that you want to deploy.

Project deployment file
 Integration Services catalog

Path:
F:\ssis\FTP\bin\Development\FTP.ispac Browse...

Enter the destination server name and where the project will be located in the Integration Services catalog.

Server name:
YOUR-SERVER-NAME Browse...

Path:
/SSISDB/Production Environment/FTP Browse...

Review your selections.

Summary:

- [-] Use the following arguments to perform this deployment from the command line:
 - [-] Command line: /Silent /SourcePath:"F:\ssis\FTP\bin\Development\FTP.ispac" /Destination
- [-] Source
 - [-] Path: F:\ssis\FTP\bin\Development\FTP.ispac
- [-] Destination
 - [-] Server name: **your-server-name**
 - [-] Path: /SSISDB/Production Environment/FTP

On the final screen after reviewing your selections click 'Deploy'.

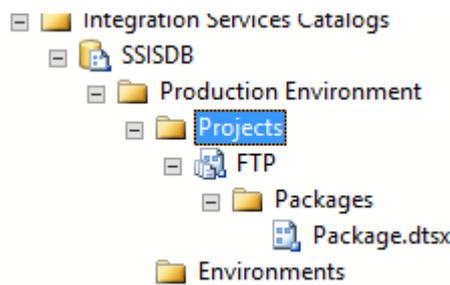
Results

Action	Result
✓ Loading project	Passed
✓ Connecting to destination server	Passed
✓ Changing protection level	Passed
✓ Deploying project	Passed

Save Report...

< Previous Next > Close Cancel

Click close. Refresh SSMS view. The SSIS Project you have just imported will be displayed under the 'Projects' folder as shown below.

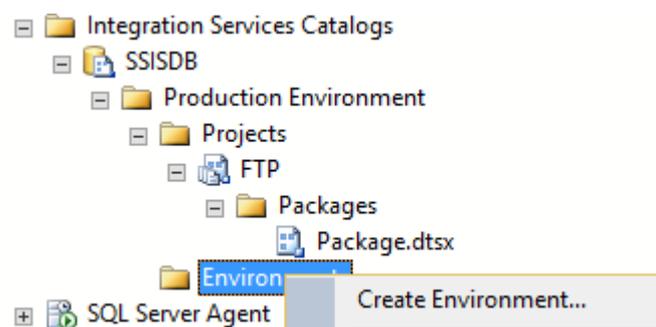


Give your environment a meaningful name, use 'Production', 'Test' or 'Dev.'

Step 4

In step 4 you will create an environment. The 'Environment' will be used to store variables. The variables will be created in step 5 and will hold values for 'FTP User name' and 'FTP Password' which will be passed to the SSIS Project when executed.

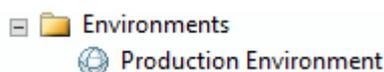
Right click Environments folder and select 'Create Environment'.



Give your environment a meaningful name, use 'Production', 'Test' or 'Dev.'



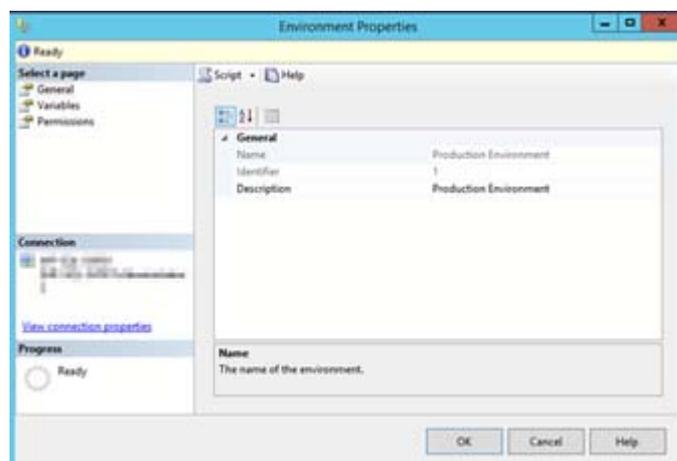
You should then see your environment listed



Step 5

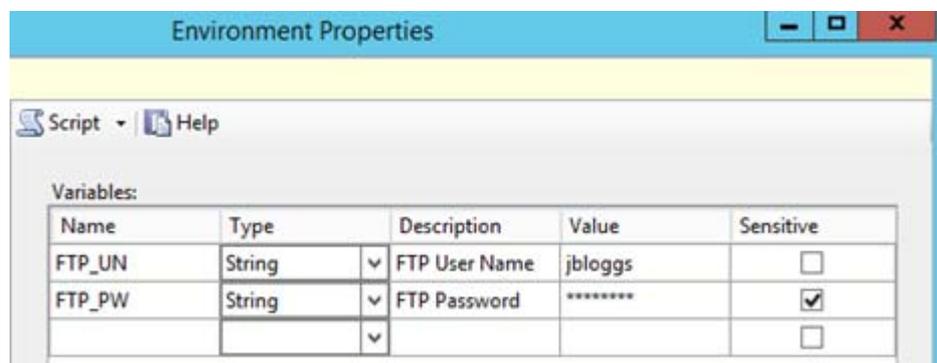
In step 5 you will create 'Environment Variables.' The variables declared here will be referenced in the SSIS Project deployed in step 3.

Right click the environment created in the previous step. Select 'Properties.' You should see this dialog.



In the 'Select a Page' pane select 'Variables'.

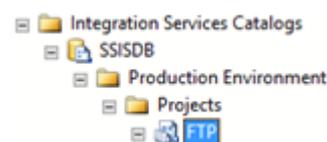
Create two new variables as shown below. One variable should contain your 'FTP Username', the second variable should contain your 'FTP Password.' Variables marked as sensitive are encrypted.



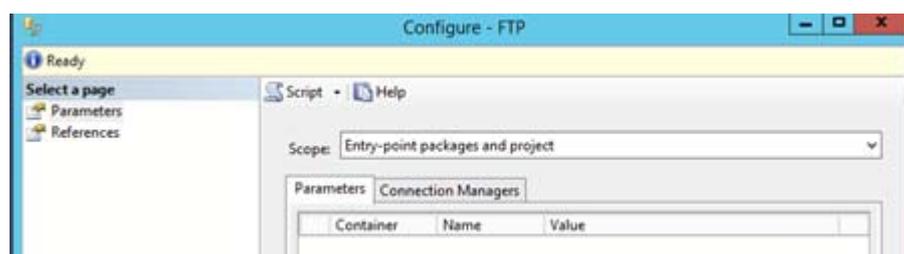
Step 6

In step 6 you will configure your project to reference the two environment variables 'FTP_UN' and 'FTP_PW' created in step 5.

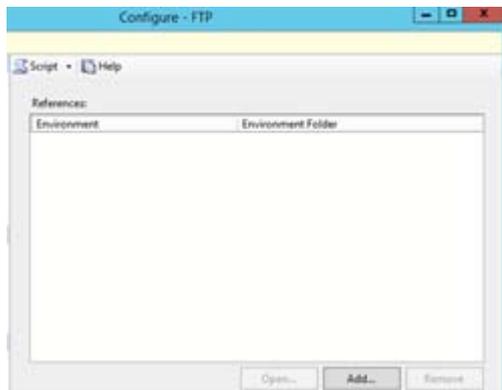
Use 'Projects, FTP.' Right click your project "FTP" then select 'Configure'.



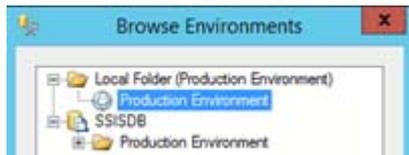
You should see the dialog below:



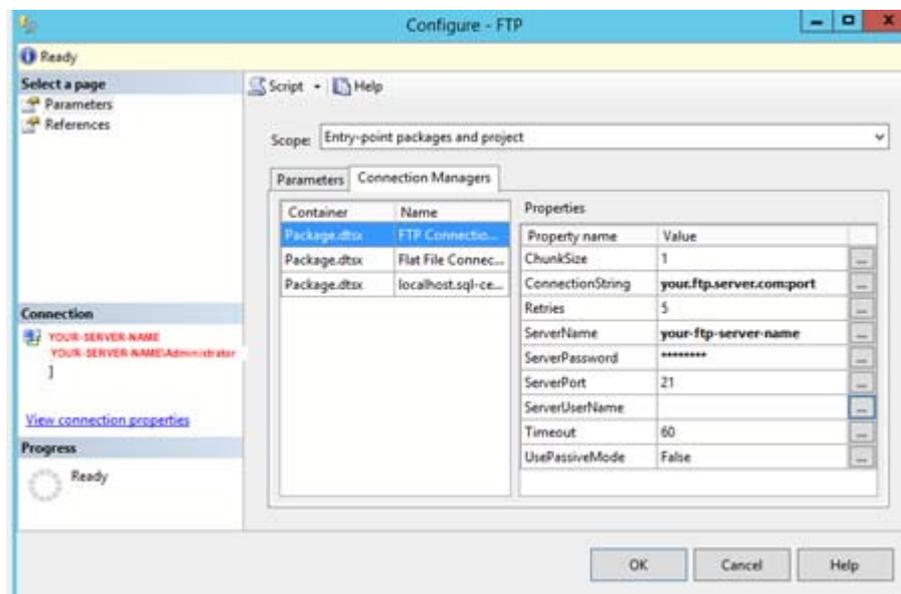
In the 'Select a page' pane select 'References'.



Click 'Add.' From the 'Browse Environments' pane select the 'Environment' created in step 4.



Select 'References' page and then the 'Connection Managers' tab.



We will now substitute values for 'ServerPassword' and 'ServerUserName' with the environment variables created in step 5. Click the ellipse next to the value for ServerPassword then select the environment variable 'FTP_PW'.



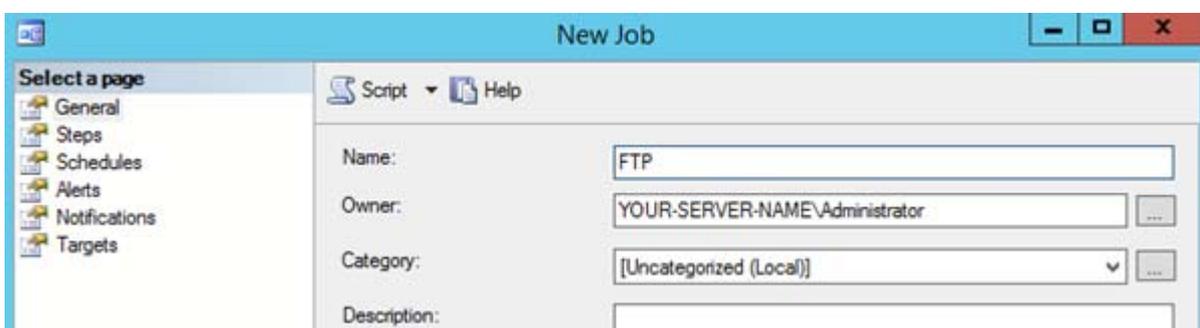
You should see your selection once you click OK.

Properties	
Property name	Value
ChunkSize	1
ConnectionString	your.ftp.server.com:port
Retries	5
ServerName	your-ftp-server-name
ServerPassword	FTP_PW
ServerPort	21
ServerUserName	
Timeout	60
UsePassiveMode	False

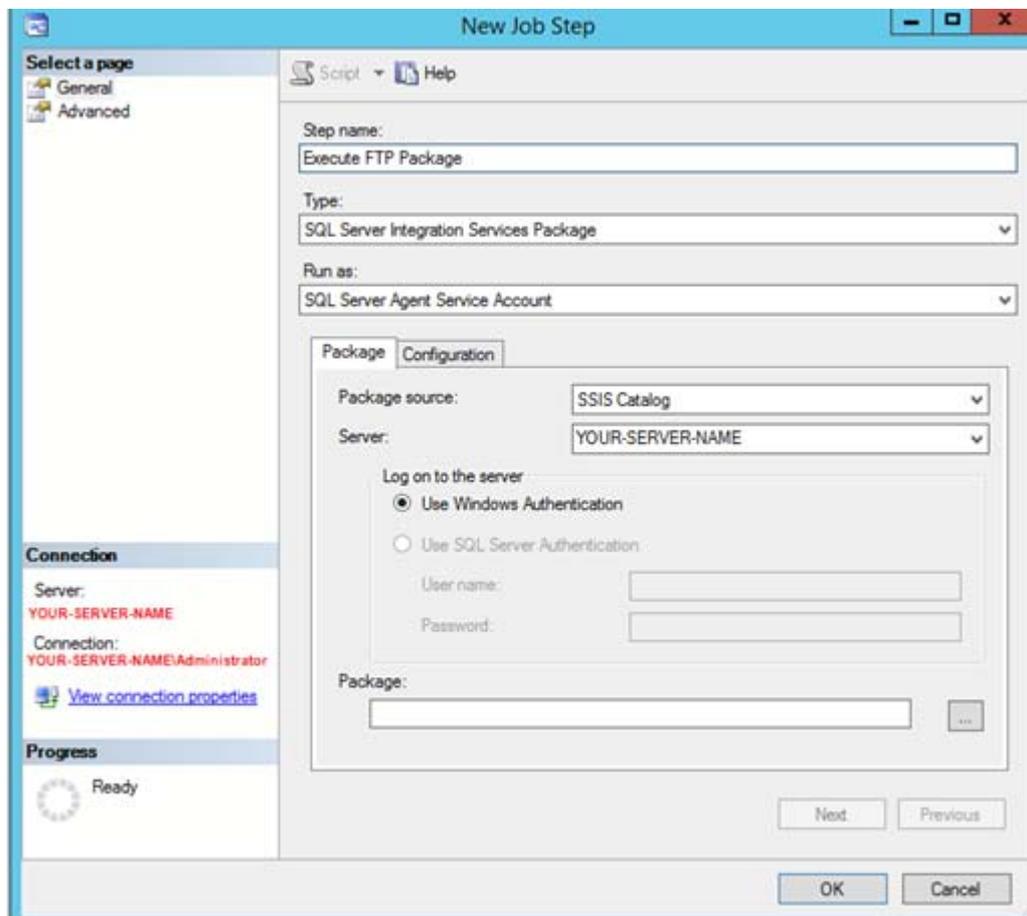
Repeat the above step for ServerUserName, selecting environment variable 'FTP_UN.'

Configure the SQL Job

The final task is to create a SQL Job and schedule it to run on a specified day/time to execute the 'SSIS Package.' Within SSMS right click SQL Server Agent then select 'New Job.' This will launch the 'New Job' wizard. Enter an appropriate name for your job. You may also wish to change the owner from the default.



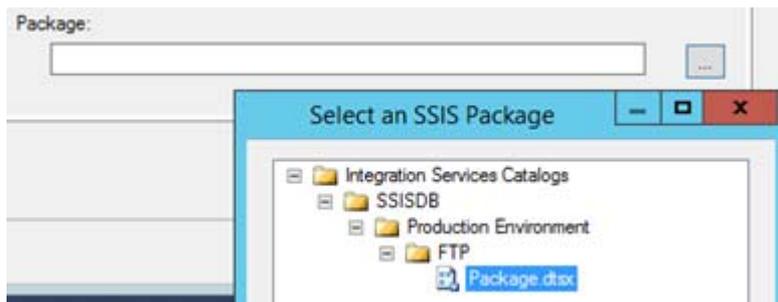
Select 'Steps'. Select 'New' to add a new step to the job. In the 'New Job Step' wizard for type select 'SQL Server Integration Services Package'. For 'Package Source' ensure 'SSIS Catalog' is selected.



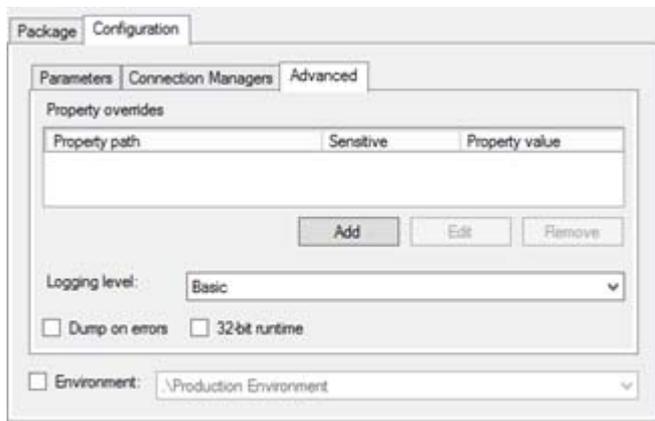
In order to run your SSIS Project deployed in step 3 click the ellipse next to 'Package.'



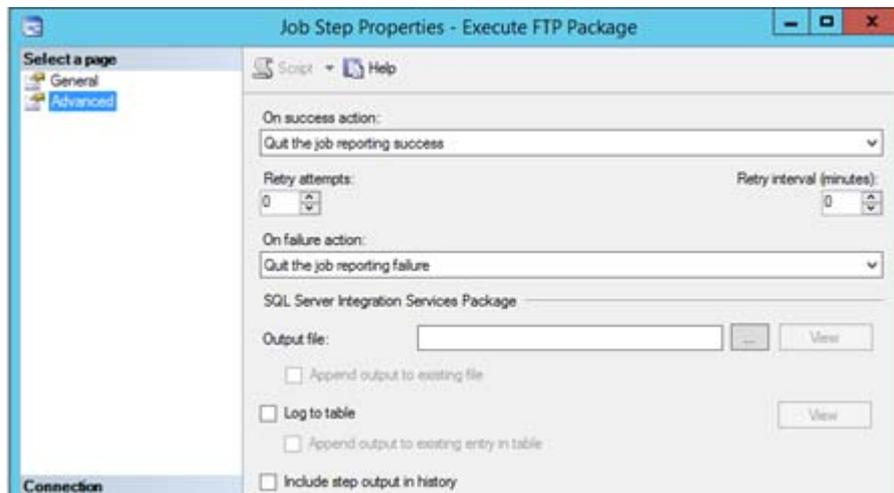
Then choose your package.



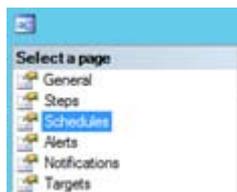
Select 'Configuration.' Select 'Advanced.'



Select the 'Environment' check box. Make sure the value matches the environment previously configured in step 4. Select the 'Advanced' option within the 'Job Step Properties'. Configure the 'On Success', 'On Failure' actions as required. Click 'OK.'



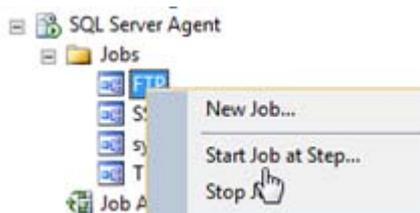
Finally within the 'Job Properties' wizard select 'Schedules'.



Select 'New' to launch the 'New Job Schedule' wizard. Give your schedule an appropriate name. Configure the job schedule to suit your environment.

Click OK and exit Job setup.

To test the job select the job name, right click and select 'Start Job at Step...'



The job will execute as shown below reporting success.



Summary

This article follows on from my first one, FTP a Dynamically Named Flat File, and explains how to create an 'Integration Services Catalog', deploy the SSIS Project created in Part 1 and then run the SSIS Project developed in Part 1 on a schedule as SQL Server Job. This is necessary to overcome the error that will occur with the FTP task.

Resources

SSIS Catalog Environments - <http://www.sqlservercentral.com/articles/Stairway+Series/121490/>

Copyright © 2002-2017 Redgate. All Rights Reserved. [Privacy Policy](#). [Terms of Use](#). [Report Abuse](#).