

Blog: Easily Switch Target Database When Executing SSIS Package with Integration Services Catalog

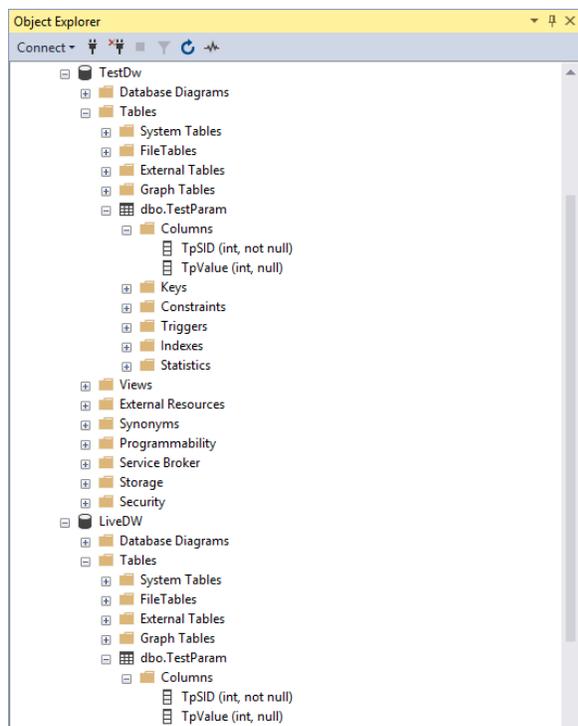
Introduction

Best practice for deploying SSIS projects is now to deploy the entire project to the Integration Services Catalog (SSIS DB Catalog), from where executions can be scheduled (!) from SQL Server Agent directly, without the need to call the dtexec utility.

In doing this you can easily pass a parameter at execution time to define which target database to load, making it easy to run packages in different modes such as test and live.

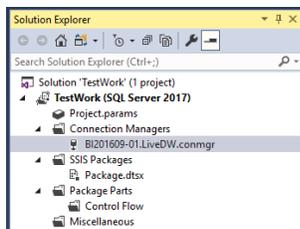
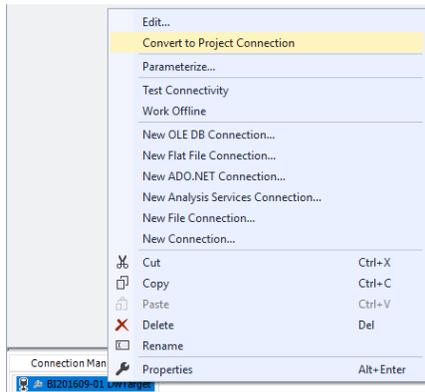
NB this can be extended by creating environments in SSIS DB but this blog describes setting a single parameter.

I have two identical target databases for this exercise, named LiveDW and TestDW, both with a single table:

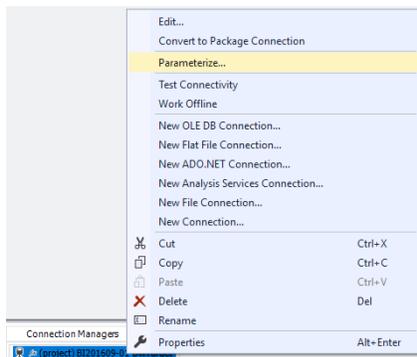


Parameterise the Connection Manager

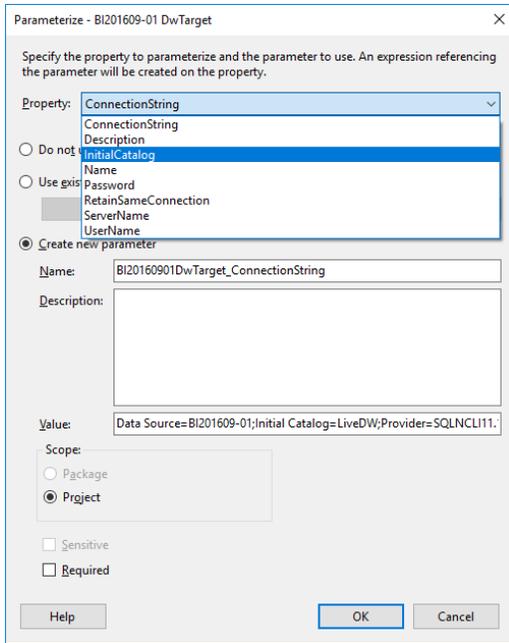
First set your target connection manager to project level:



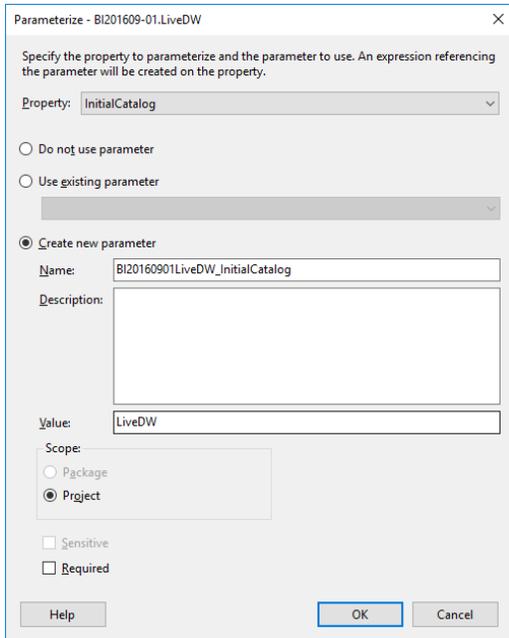
Then parameterise the (now project level) connection manager:



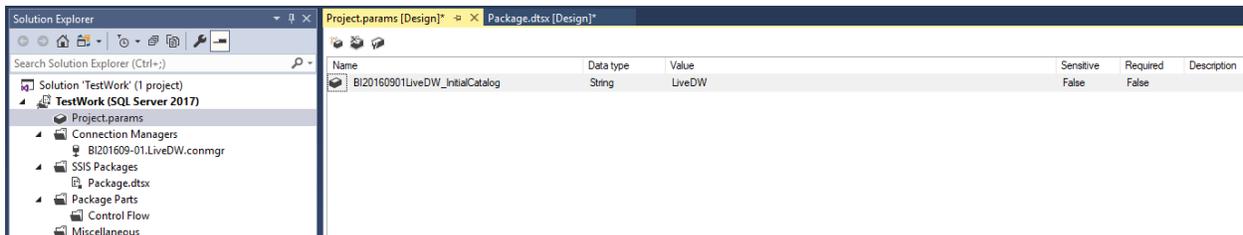
In my case the Test and Live target databases are both on the same SQL Server instance accessible by a single user so I just need to parameterise the initial catalog:



It populates with the current value:



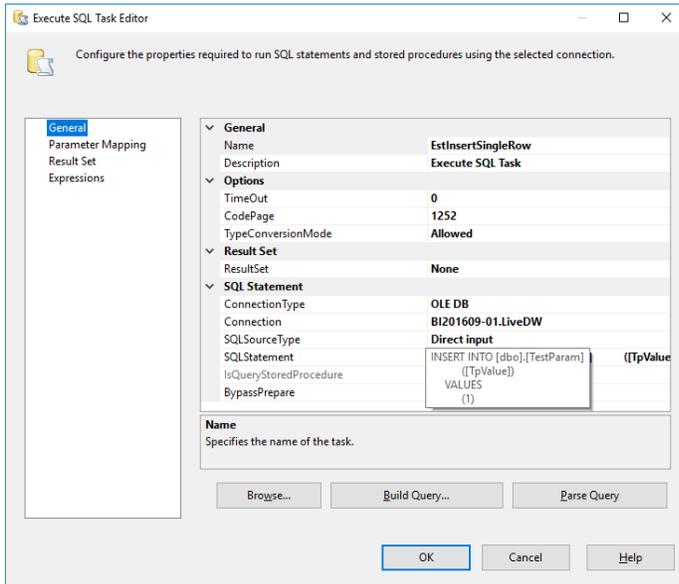
Double click Project.params under your project in Solution Explorer to see the parameter:



Changing the text in the Value field to either LiveDW or TestDW determines which database will be loaded when executing the job from within Visual Studio (and provides the default value when deploying to SSIS DB Catalog).

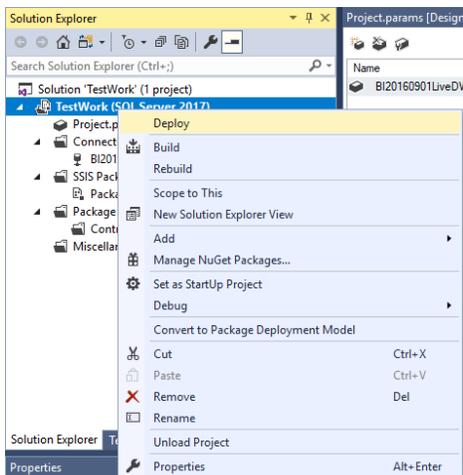
Add a Task

My test job has a single Execute SQL Task which inserts a single row into the target database



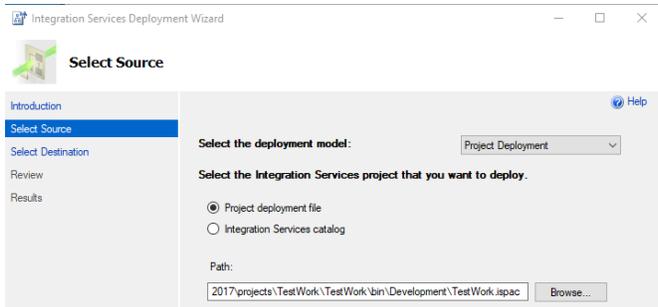
Deploy the Project to Integration Services Catalog

To deploy the project to the SSIS DB Catalog (which you need to create in SSMS if it does not already exist on your SQL Server instance and create a folder to deploy your project into), right click the project and select Deploy:

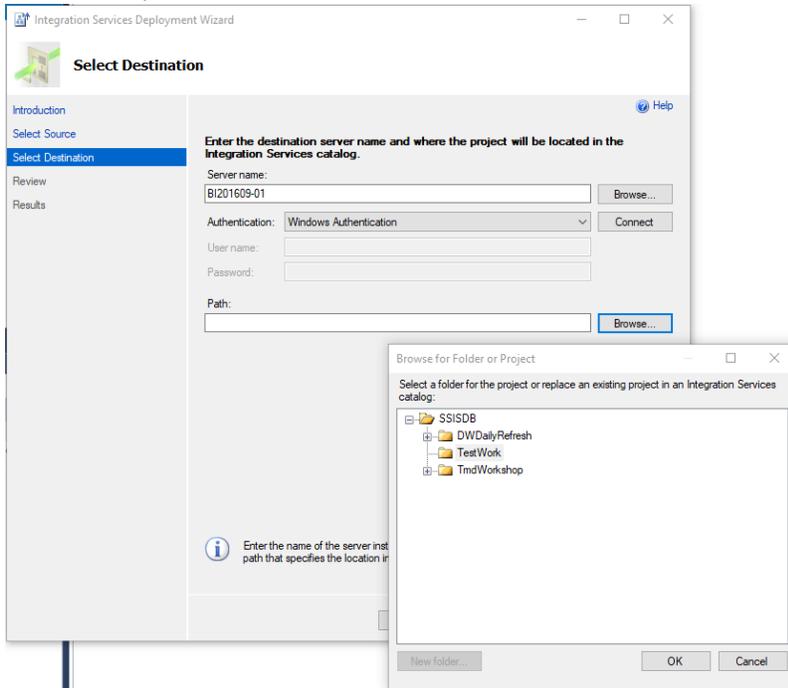


This will walk you through a wizard to deploy your code.

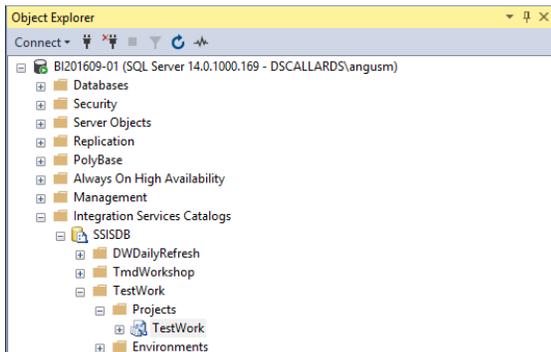
Select Project Deployment as the deployment mode, the path will default to the file path to your project's .ispac file:



Connect to your server then browse to the relevant folder in the SSIS DB:

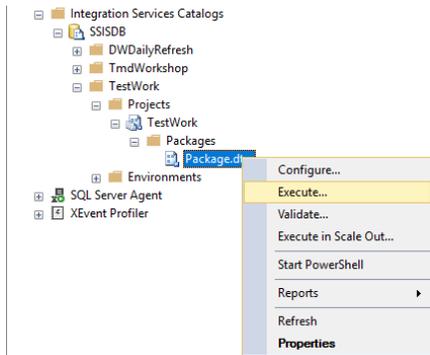


Review, Deploy and Close. Now switching into SSMS you should see your project under the SSIS DB Catalog:

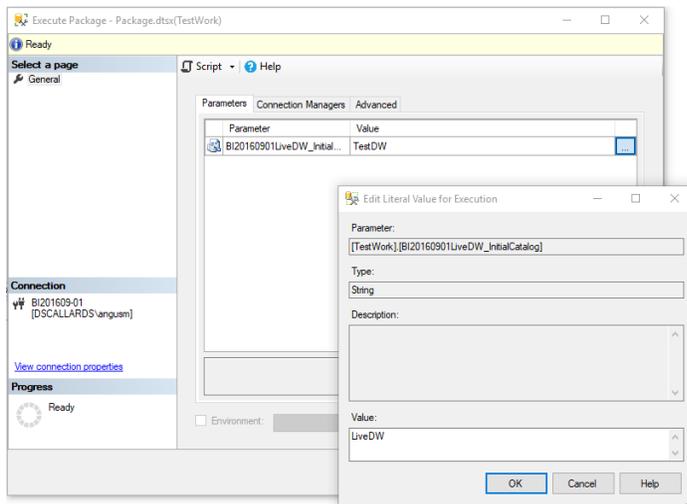


Test a Package

Expand the project to see the packages within it (in this only one). To test the package you can execute from here:



On the Parameters tab, click the ellipsis next to Value to change the parameter value:



Click OK to run and again to view the execution report:

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the server hierarchy for BI201609-01. The main pane shows the 'Overview' page for a package task. The 'Execution Information' section provides details about the operation, including the package name 'TestWork\TestWork\Package.dtsx', environment, status 'Succeeded', and machine 'BI201609-01'. The 'Execution Overview' table shows the results of the package execution, with all tasks succeeding. The 'Parameters Used' table lists various parameters such as 'LiveDW', 'DUMP_ON_ERROR', and 'LOGGING_LEVEL' with their respective values and data types.

Operation ID	10	Duration (sec)	2.388
Package	TestWork\TestWork\Package.dtsx	Start Time	31/07/2019 17:15:32
Environment	-	End Time	31/07/2019 17:15:35
Status	Succeeded	Caller	DSCALLARDS\angum
Machine	BI201609-01		

Result	Duration (sec)	Package Name	Task Name	Execution Path
Succeeded	0.14	Package.dtsx	Package	.\Package
Succeeded	0.062	Package.dtsx	EstInsertSingleRow	.\Package\EstInsertSingleRow

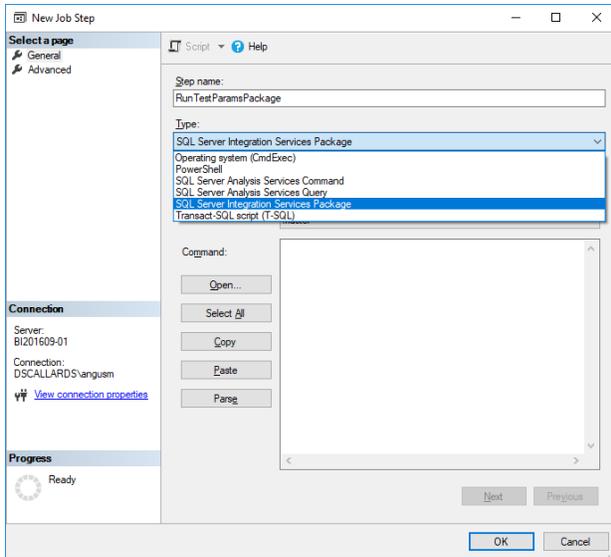
Name	Value	Data Type
BI20160901LiveDW_InitiaC_atalog	LiveDW	String
CALLER_INFO		String
DUMP_EVENT_CODE	0	String
DUMP_ON_ERROR	False	Boolean
DUMP_ON_EVENT	False	Boolean
LOGGING_LEVEL	1	Int32
SYNCHRONIZED	False	Boolean

Schedule the Package

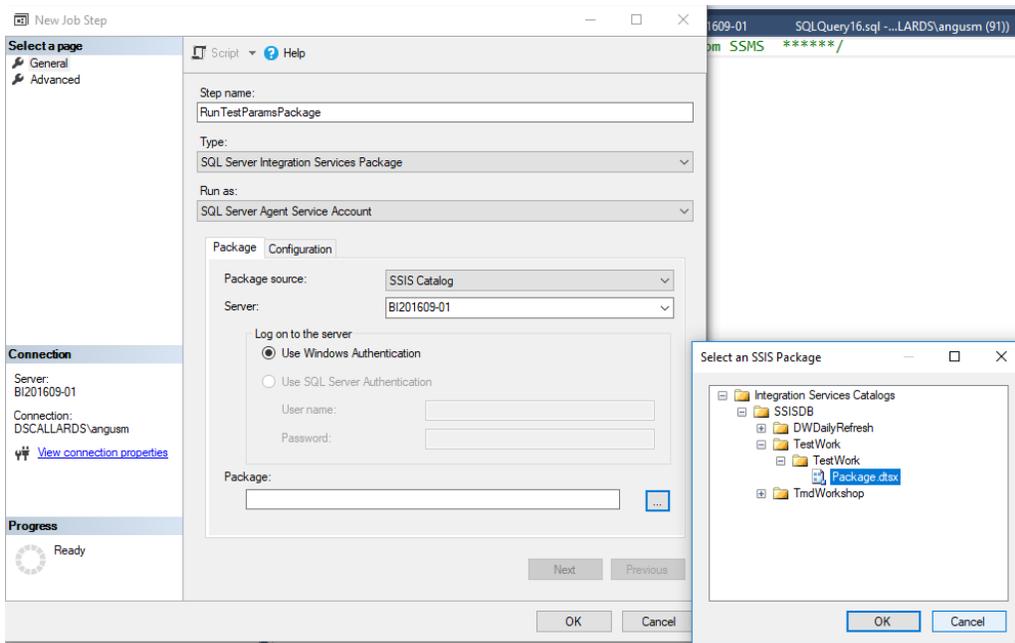
Now to schedule the package, go to the SQL Server Agent node and select New Job:

The screenshot shows the SQL Server Enterprise Manager interface with the 'Jobs' folder selected under the 'SQL Server Agent' node. A context menu is open over the 'Jobs' folder, showing options such as 'New Job...', 'Manage Schedules', 'Manage Job Categories', 'View History', 'Filter', 'Start PowerShell', 'Reports', and 'Refresh'.

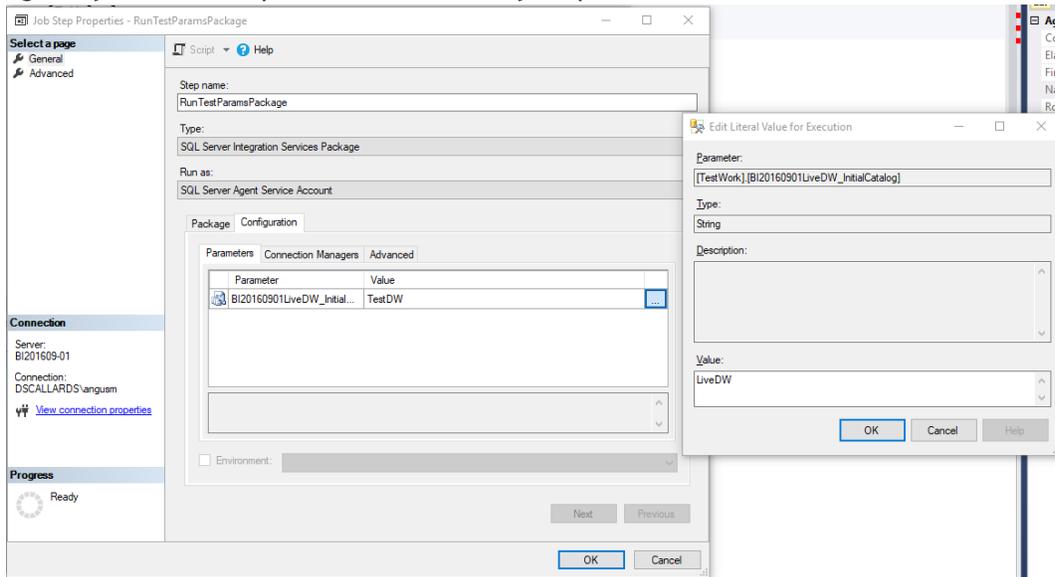
In the General page, provide a name for the job then on the Steps page click the New button, specify a name for the step and a Type of SQL Server Integration Services Package



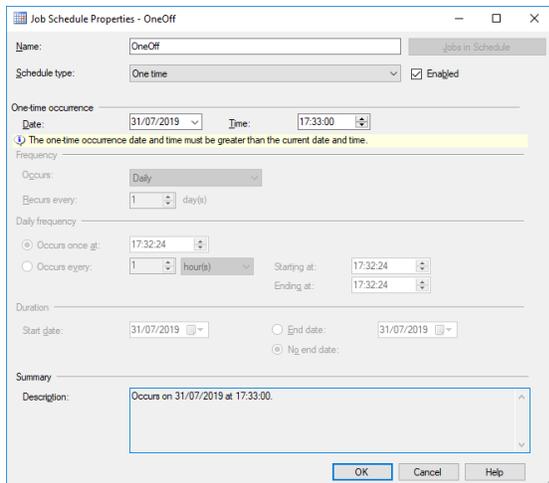
Enter the package source as SSIS Catalog, your server name then click the ellipsis to the right of the Package field to select your package:



Then go to the Configuration tab which is where you set the parameter value for the target database, again by click the ellipsis next to Value on your parameter:



Then on the schedule page, schedule the job as normal:



You should see the data appear in the relevant database

Angus Menter, BI Practice Manager
DSCallards Ltd