SentryOne Test Visual Studio Extension Data-Driven Testing

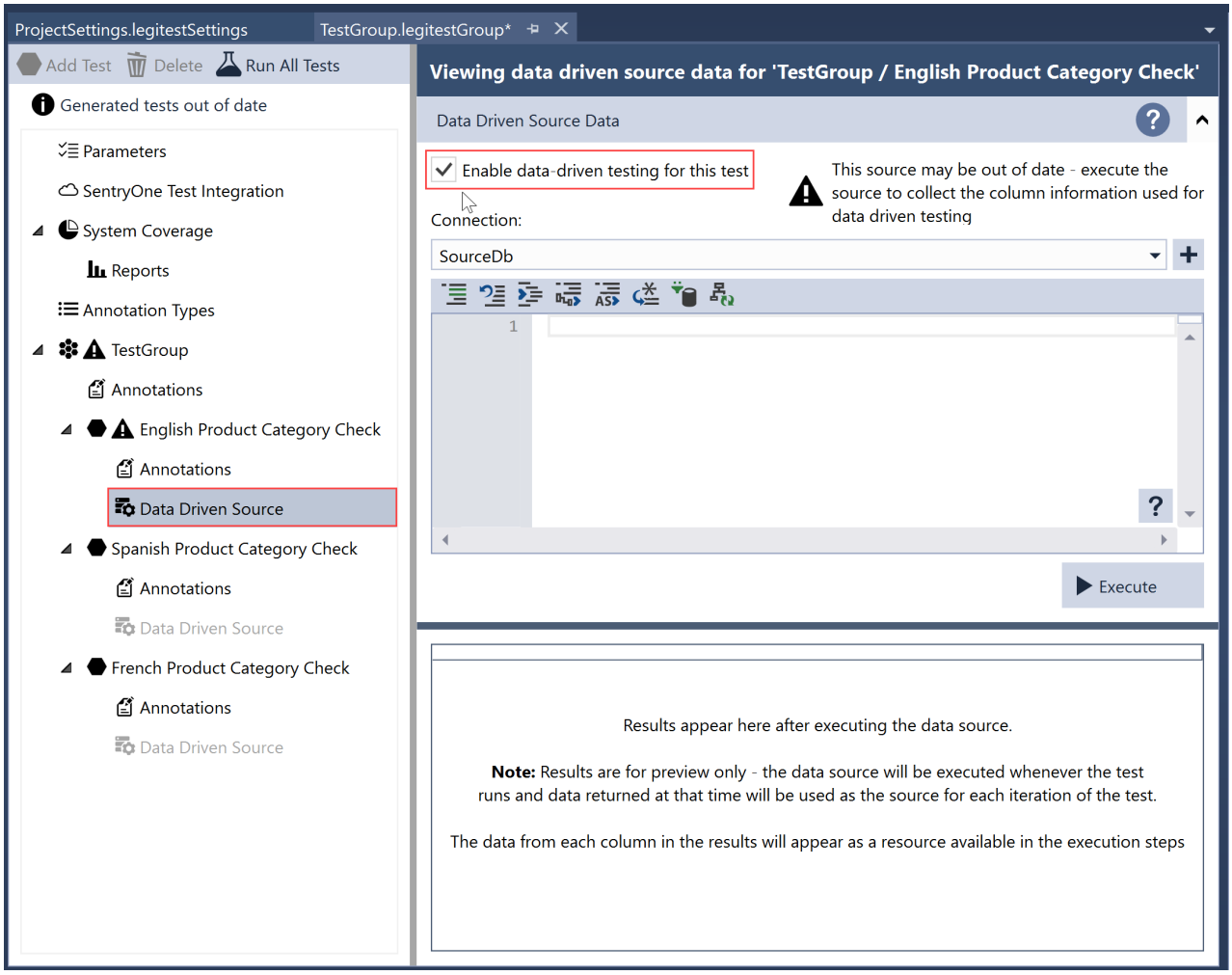
Last Modified on 13 November 2019

Data-driven testing allows you to repeat the same test many times, based on dynamic source data. For example, consider the following scenario:

You want to ensure that each of a series of 2000 products passes a set of tests, and you don't want to create 2000 of the same test. Additionally, it might not be possible to write a single query that covers all the test cases (you may want to validate that the results of a REST call for each of the products passes a set of criteria). This is where data-driven testing comes in to play.

Configuring data-driven testing

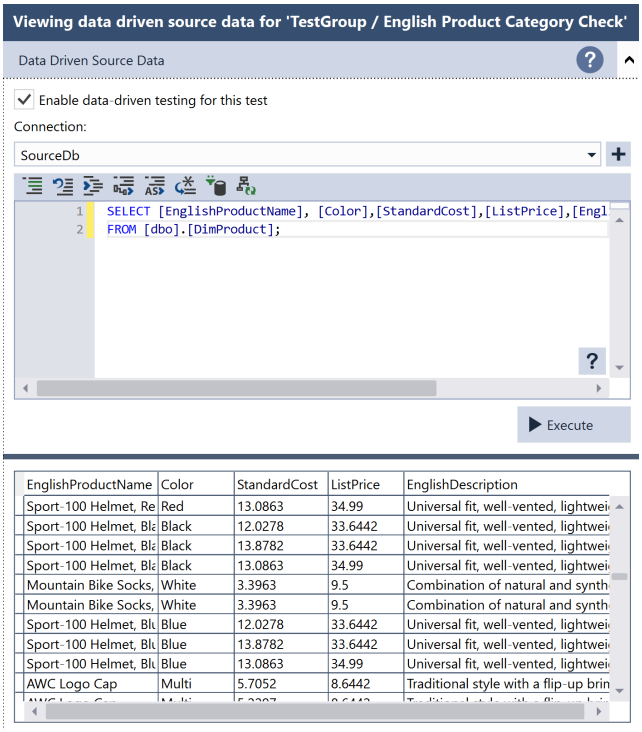
In the test tree on the left, under each test node is a **Data Driven Source** node. Select the desired **Data Driven Source** node, and then select the **Enable data-driven testing for this test** check-box to display the following:



Here, much like in the Interactive Comparison Wizard you are selecting a connection and entering a query or configuring the source (if using a REST connection, for example). Once you have selected the connection, select **Execute** to display the results.

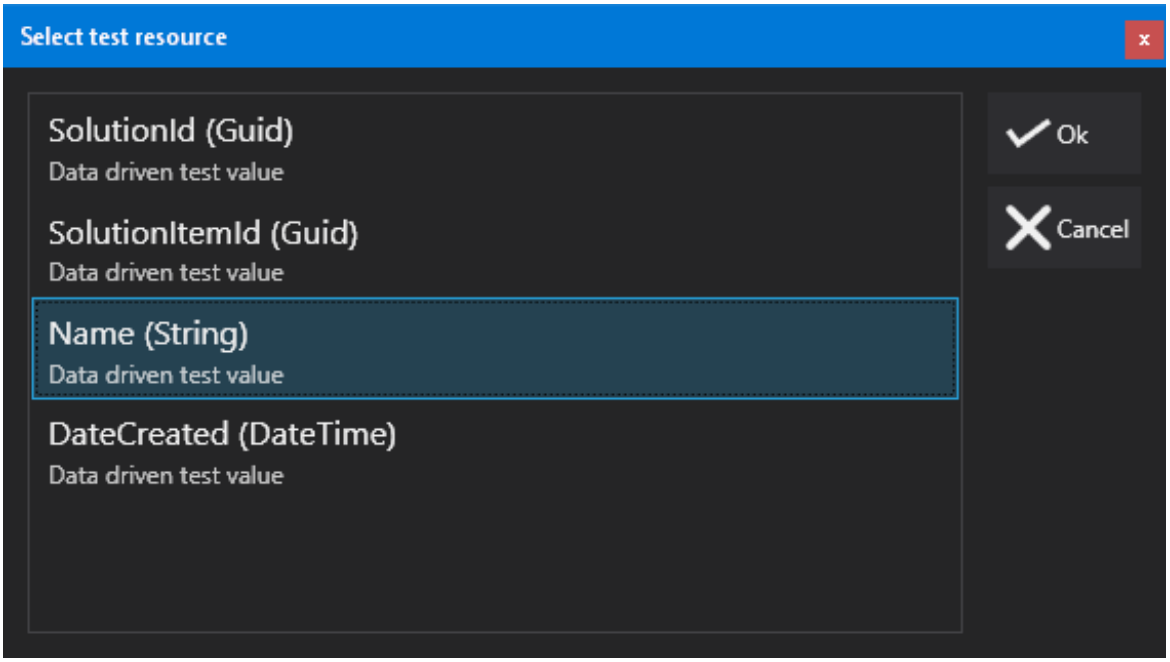
> ⚠ **Important:** The results are shown in the UI to enable easier configuration. The data returned is not stored, and the source is run every time the test runs. The only information that's retained is the metadata about the columns, so that they can be used as resources.

One you've configured the source and executed it, you can see that the data is shown (this example is using a list of solution items from the DOC xPress metabase):



Success: This test is now configured for data-driven testing.

Now, when selecting resources for actions or assertions, the columns returned from the Data Driven Source are reflected in the list of available resources:



Use these values wherever you would use parameters normally. For example, you could write a query in the following form:

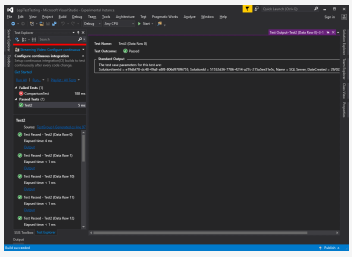
```
SELECT COUNT(*) FROM LineageService.ItemSets WHERE SolutionItemId = ''
```

Differences between data-driven testing in MS Test and NUnit

There are differences in how data driven testing works between the test frameworks available. For a complete description of the differences, see framework considerations below.

Framework Considerations

Data-driven testing is implemented differently under MS Test and NUnit. Both evaluate the data-driven source during the discovery phase of the test and use the output of that to generate the list of iterations. However, there are some differences in the output that displays from each because the test frameworks are designed differently.

Framework	Description	Example
MSTest	When using MSTest each iteration of the test is reported under a single test in the test explorer. When looking at the output for this test, you see several different runs. Each one has a link that says **output** that can be selected to get a summary of the rows of data used for that test. When examining the output of an individual test, the display looks similar to this example. **Note:** MS Test data driven tests generate an XML file containing the test cases during the discovery phase. This is then used as the source of the test data using the **DataSource** attribute.	
	When using NUnit each iteration of the test is reported as an individual test.	

Framework	Description	Example
NUnit	This can make it a lot easier to identify the tests. Here is a similar display, where we are looking at the output of an individual test iteration and viewing the output. **Note:** The technical implementation of the NUnit back end is slightly cleaner than MSTest because NUnit supports the TestCaseSource attribute meaning that the test data doesn't have to be written to disk first.