Create a Data-Driven Unit Test (*CoolMath project*)

Visual Studio 2015

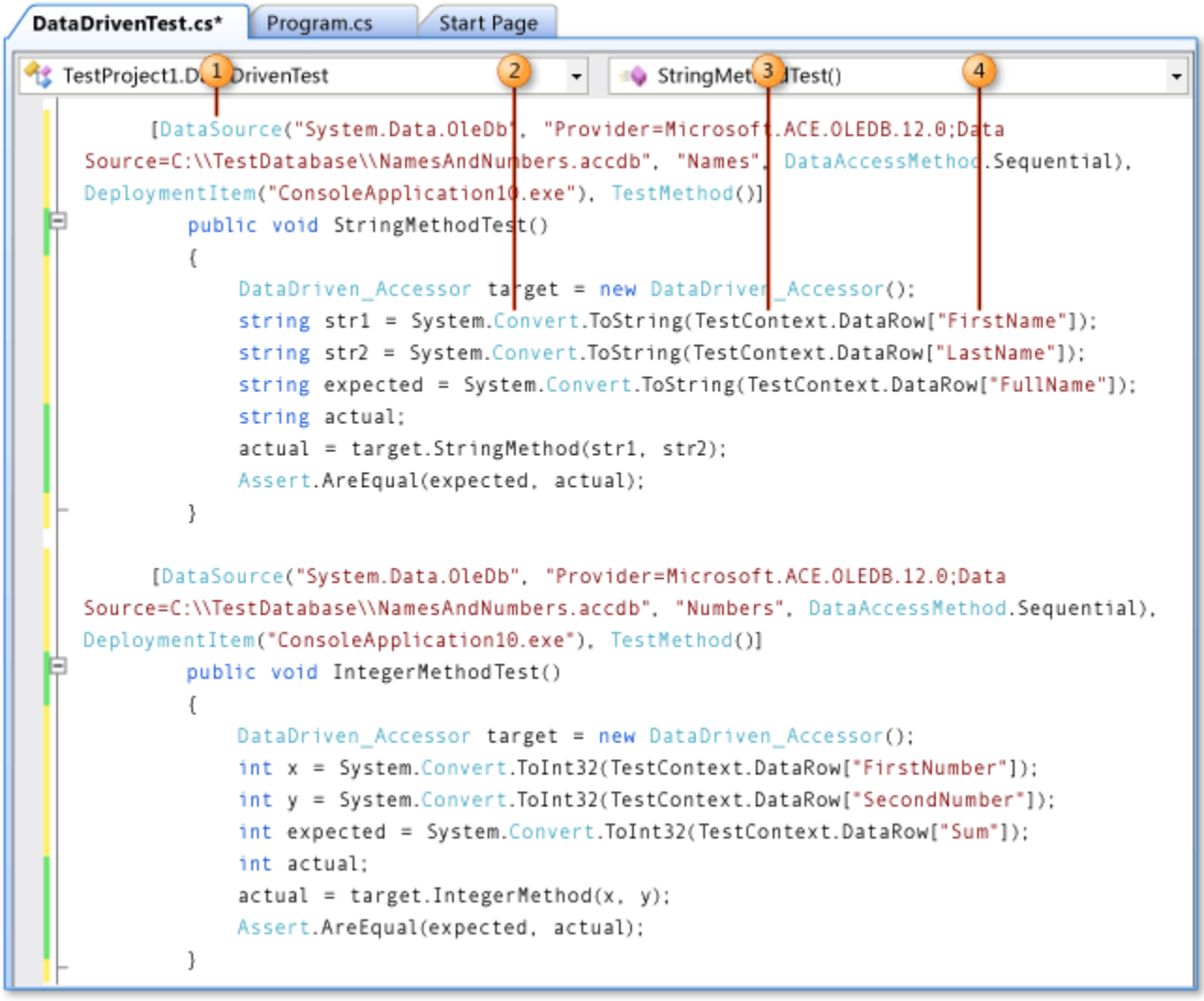
You can set up a unit test so that instead of typing values into a test method, you can retrieve the values from a data source. The unit test is run successively for each row in the data source. This makes it easy to test a variety of input in a single test run.

There are two phases to setting up a unit test to retrieve values from a data source. The first is to make the connection between the unit test method and the data source by using the Properties of the unit test. The second phase is to assign the method's variables to retrieve their values from the appropriate column in your data source. You do this in the logic of the test method itself.

Example of a Data-Driven Unit Test

The following figure shows two unit tests: one adds two numbers and the other concatenates a first name with a last name. These unit tests are set up to retrieve the values for the tests from a database.

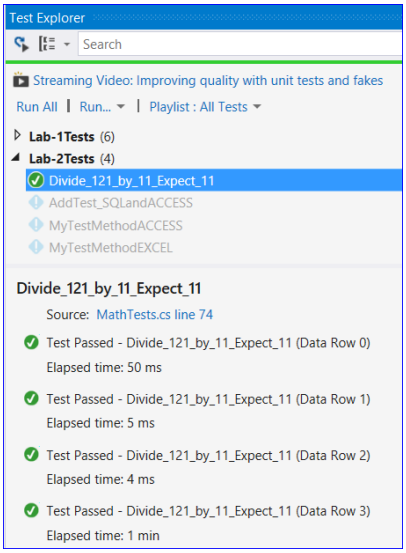
Data-driven unit tests



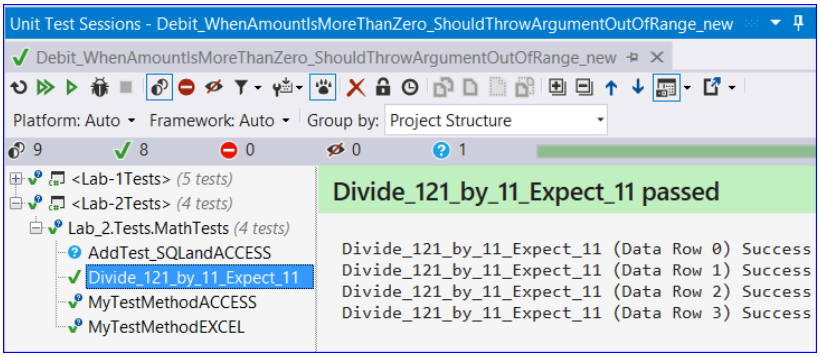
1. This line provides the *DataSource* attribute and the connection string. The exact information in the connection string differs, depending on what kind of data source you are using. In this example, we used an Access database.
2. When the values come in from the database table, they must be converted to the appropriate type. In this example, the first unit test converted the values from the database into strings and the second unit test converted them to integers.
3. The TestContext.*DataRow* property tells the unit test which row to retrieve data from. In this case it will retrieve data starting from the first row and then sequentially until it reaches the last row in the table. For more information about this property, see *TestContext*.
4. The name of the column tells the unit test which column to retrieve data from.

You can see the detailed results for each row of data that the test used by double-clicking the test in the **Test Results** window.

The detailed results of a data-driven unit test



or



Make a connection between your unit test and your data source

This is the first phase of setting up a unit test to use a data source. In this phase, you make the connection between the unit test method and the data source.

This procedure shows how to use the Properties of the unit test to create the connection. However, you can also create the connection by creating an ***app.config*** file and adding connection information to it. The advantage to using an ***app.config*** file is that you can change the location of the database without making changes to the unit test itself. For information about how to create and use an ***app.config*** file, see Walkthrough: Using a Configuration File to Define a Data Source.

Note

If you are familiar with data connection strings, you can type the data connection string after the first bracket of the [TestMethod()] element instead of using the properties window.

[*DataSource*(data connection string goes here), *TestMethod*()]

Assign variables to take their values from your data source

This is the second phase of setting up a unit test to use a data source. In this phase, you assign the method's variables to retrieve their values from the appropriate column in your data source.

To assign variables to values from your data source

1. Open the unit test file that contains the test method for which you want to use a data source and locate the variables in the test method.
2. For each variable that you want to come from the data source, use the syntax *TestContext.DataRow*["NameOfColumn"].

> **Note**
>
> You might have to convert the data types from those of the data source to those of your test code project, as shown in the previous example.

Run the unit test and view results

You run a data-driven unit test just as you would any other unit test. You can view the detailed results for each row of data that the test used by double-clicking the test in the **Test Results** window.

To run and view the results of a data-driven unit test

1. Right-click anywhere in your unit test and then click **Run Tests**.
2. After the test runs, double-click the test in the **Test Results** window to view the results of each test iteration in the **Data Driven Test Results** window.

See Also

Reference

TestContext