


## **SilkTest 13.0**

---



**SilkTest Classic -  
Creating Data-  
Driven Test Cases**

**Micro Focus**  
575 Anton Blvd., Suite 510  
Costa Mesa, CA 92626

Copyright © 2012 Micro Focus. All rights reserved. Portions Copyright © 1992-2009 Borland Software Corporation (a Micro Focus company).

**MICRO FOCUS**, the Micro Focus logo, and Micro Focus product names are trademarks or registered trademarks of Micro Focus IP Development Limited or its subsidiaries or affiliated companies in the United States, United Kingdom, and other countries.

**BORLAND**, the Borland logo, and Borland product names are trademarks or registered trademarks of Borland Software Corporation or its subsidiaries or affiliated companies in the United States, United Kingdom, and other countries.

All other marks are the property of their respective owners.

2012-05-03

# Contents

<b>Overview of Data-Driven Test Cases</b>	<b>4</b>
<b>Before You Begin</b>	<b>5</b>
<b>Creating a Data-Driven Test Case and Running It Using the Sample Record</b>	<b>6</b>
Creating a Data-Driven Test Case	6
Running Your Data-Driven Test Case	7
<b>Running a Data-Driven Test Case Using All Rows and Tables</b>	<b>8</b>
Creating a Data-Driven Test Case	8
Find and Replace Values	9
Running Your Data-Driven Test Case	9

# Overview of Data-Driven Test Cases

A data-driven test case lets you store data combinations in a list of items and invoke the test case once for each item, passing the data to the test case as a parameter. You can think of a data-driven test case as a template for a class of test cases. The benefits are easy to see:

- Data-driven test cases reduce redundancy in a test plan.
- Writing a single test case for a group of similar test cases makes it easier for you to maintain scripts.
- Data-driven test cases are reusable; adding new tests only requires adding new data.

You can use the **Data Driven Workflow** to help you create data-driven test cases that use data stored in databases. Previous versions of SilkTest Classic explained how to store data in a test plan, how to type in the data at the time you are running the test case, and how to pass in data from an external file. While those techniques still work, we strongly recommend that you now use the **Data Driven Workflow** which generates much of the necessary code for you and guides you through the required steps.

Regardless of the technique you use, the basic process for creating a data-driven test case is:

1. Create a standard test case. It will be very helpful to have a good idea of what you are going to test and how to perform the verification.
2. Identify the data in the test case and the 4Test data types needed to store this data.
3. Modify the test case to use variables instead of hard data.
4. Modify the test case to specify input arguments to be used to pass in the data. Replace the hard coded data in the test case with variables.
5. Call the test case and pass in the data, using one of four different techniques:
  - Use a database and the **Data Driven Workflow** to run the test case, the preferred method.
  - Click **Run > Testcase** and type the data in the **Run Testcase** dialog box.
  - In a QA Organizer test plan, insert the data as an attribute to a test description.
  - If the data exists in an external file, write a function to read the file and use a `main()` function to run the test case.

# Before You Begin

Before you start working with data-driven test cases, complete the following tasks:

- Get the files needed by the exercise. To do this, go to your <SilkTest installation directory>\examples directory and unzip the `ddw.zip` file into that same directory. The following files should now be in your <SilkTest installation directory>\examples directory:
  - `frame.inc`
  - `gentest1_example.g.t`
  - `gentest1_example.res`
  - `gentest2_example.g.t`
  - `gentest2_example.res`
  - `testexcel.xls`
  - `usability.t`
- Take a look at the Microsoft Excel file, which is used as the database by this exercise, by opening `testexcel.xls` located in the <SilkTest installation directory>\examples directory. If you cannot find this file in this directory, you have not unzipped the files as described in the preceding bulleted step.



**Note:** The following three worksheets are included in `testexcel.xls`:

- Products
- Customers
- CreditCards

Each of these worksheets contains several columns of data. For example, on the Products worksheet, there are columns named Desc, Blurb, NumInStock, QtyToOrder, and OnSale.

- Ensure that you have the correct DSN available: Silk DDA Excel. This is automatically set up when you install SilkTest. On Windows, for example, click **Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC)**. Click the **System DSN** tab and verify that the Silk DDA Excel driver is listed. This is the default DSN used by the Data Driven Workflow and, coincidentally, the one used by these exercises.
- If you are not familiar with the GMO application, navigate around the application to become familiar with it. You can locate this application at: <http://demo.borland.com/gmopost/>.
- Click **Options > Runtime** to open the **Runtime Options** dialog box. In the **Use files** text box type `extend\explorer.inc`. Then click **Options > Extensions** and enable the primary extensions for your browser.

# Creating a Data-Driven Test Case and Running It Using the Sample Record

When you create a data-driven test case, one of the things that SilkTest Classic does is to write a sample record for each table. SilkTest Classic does this by inserting a value from the original test case into the sample record. You can then use the sample record to run a quick test of your data-driven test case or to verify your DSN connection.

The goal of this exercise is for you to use the Data Driven Workflow to create a data-driven test case and to run that test case using the sample record.

## Creating a Data-Driven Test Case

1. Open the GMO application at <http://demo.borland.com/gmopost/>.
2. Start SilkTest Classic and click **File > Open**.
3. Browse to `<SilkTest installation directory>\examples` and open `usability.t`.
4. Click **Workflows > Data Driven** to turn on the Data Driven Workflow bar, if it's not already visible.
5. On the Data Driven Workflow bar, click **Data Drive Testcase**.
6. On the **Select Testcase** dialog box, double-click `testcase1`.
7. On the **Specify Data Driven Script** dialog box, click **Create a new file/Overwrite an existing file** and browse to the `<SilkTest installation directory>\examples` directory. Give the file the name `gentest1`, which means you will have to change the default new name.
8. Click **Save** to save the new file and then click **OK** when you return to the **Specify Data Driven Script** dialog box. SilkTest Classic automatically applies the `.g.t` file extension to that file.
9. On the **Select Data Source** dialog box, specify the location of the database you are using to supply values for the data-driven test case. Accept the default data source, **Silk DDA Excel**. At the **Select Workbook** field, click **Browse** and select the `<SilkTest installation directory>\examples\TestExcel.xls` and click **Open**. Click **OK** on the **Select Data Source** dialog.
10. At the **Select Workbook** text box, click **Browse**, select the `<SilkTest installation directory>\examples\TestExcel.xls`, and click **Open**.
11. Click **OK** on the **Select Data Source** dialog box.
12. On the **Specify Data Driven Testcase** dialog box, click **OK** to indicate that you want to add a new test case named `DD_testcase1` to the `gentest1.g.t` file you have created.
13. The **Find/Replace Values** dialog box opens; close this dialog box because you are not using it for this exercise.

SilkTest Classic loads the database tables and columns and then displays the `gentest1.g.t` file in the 4Test Editor.



**Note:** There is a section at the top of your file that is collapsed; it contains the code that SilkTest Classic writes including the DSN connection, a sample record, and other necessary information:

```
[+] // *** DATA DRIVEN ASSISTANT Section (!! DO NOT REMOVE !!) ***
```

To see the code that SilkTest Classic creates, click `[+]` to expand that section. Do not edit any of the information contained in this section.

## Running Your Data-Driven Test Case

1. Click **Run Testcase** on the Data Driven workflow bar to run this test case.
2. Select **DD\_testcase1** and check **Use Sample Data from Script**.
3. Click **Run**.

Testcase1 should have run only once and the sample data from `REC_Products_ grTest_Products_` been passed into it. Compare your file with the `gentest1_example.g.t` and `gentest1_example.res` found in the `<SilkTest installation directory>\examples` directory.

# Running a Data-Driven Test Case Using All Rows and Tables

Values are text strings, numbers, and booleans (true/false) that exist in your original test cases. One of the steps in creating a data-driven test case is to find these values and replace them with references to columns in your data source.

SilkTest Classic checks to make sure that each value you selected is appropriate for replacement by the column in your test case. Click **Edit > Data Driven > Validate Replacements** while you are in a .g.t file to turn off this validation. This means that the “Find” aspect of Find and Replace works as usual, but that the values you replace are not validated. By turning off this checking, you suppress the error messages that SilkTest Classic would have otherwise displayed. Any 4Test identifier or fragment of a string is considered an invalid value for replacement unless **Validate Replacements** is turned off.

If you are new to creating data-driven test cases, we recommend that you keep this validation turned on.

The goal of this exercise is for you to use the Data Driven Workflow to:

- Create a data-driven test case.
- Find and replace values in a test case with links to the data source.
- Run the data-driven test case with all rows and tables from the data source.

## Creating a Data-Driven Test Case

1. Open the GMO application at <http://demo.borland.com/gmopost/>.
2. Start SilkTest Classic and click **File > Open**.
3. Browse to `<SilkTest installation directory>\examples` and open `usability.t`.
4. Click **Workflows > Data Driven** to turn on the Data Driven Workflow bar, if it's not already visible.
5. On the Data Driven Workflow bar, click **Data Drive Testcase**.
6. On the **Select Testcase** dialog box, double-click `testcase1`.
7. On the **Specify Data Driven Script** dialog box, click **Create a new file/Overwrite an existing file** and browse to the `<SilkTest installation directory>\examples` directory. Give the file the name `gentest1`, which means you will have to change the default new name.
8. Click **Save** to save the new file and then click **OK** when you return to the **Specify Data Driven Script** dialog box. SilkTest Classic automatically applies the .g.t file extension to that file.
9. On the **Select Data Source** dialog box, specify the location of the database you are using to supply values for the data-driven test case. Accept the default data source, **Silk DDA Excel**. At the **Select Workbook** field, click **Browse** and select the `<SilkTest installation directory>\examples\TestExcel.xls` and click **Open**. Click **OK** on the **Select Data Source** dialog.
10. At the **Select Workbook** text box, click **Browse**, select the `<SilkTest installation directory>\examples\TestExcel.xls`, and click **Open**.
11. Click **OK** on the **Select Data Source** dialog box.
12. On the **Specify Data Driven Testcase** dialog box, click **OK** to indicate that you want to add a new test case named `DD_testcase1` to the `gentest1.g.t` file you have created.
13. The **Find/Replace Values** dialog box opens; close this dialog box because you are not using it for this exercise.

SilkTest Classic loads the database tables and columns and then displays the `gentest1.g.t` file in the 4Test Editor.





**Note:** There is a section at the top of your file that is collapsed; it contains the code that SilkTest Classic writes including the DSN connection, a sample record, and other necessary information:

```
[+] // *** DATA DRIVEN ASSISTANT Section (!! DO NOT REMOVE !!) ***
```

To see the code that SilkTest Classic creates, click [+] to expand that section. Do not edit any of the information contained in this section.

## Find and Replace Values

1. Indicate the names of the table and column that contain values that you want to link to.  
You can do that with either the **Find/Replace Values** dialog box or using the **Find and Replace** menu bar; this exercise uses the **Find/Replace Values** dialog box.
2. On the **Find/Replace Values** dialog box, select Customer\$ from the list of tables and Address from the list of columns. Click **Find Next** to start searching for addresses within the test case. SilkTest Classic finds and highlights various values.
3. Click **Find Next** until SilkTest Classic highlights `201 Spring street`. Since this is an address, click **Replace** on the **Find/Replace Values** dialog box to replace that number with a value from the data source.
4. There is only one address in this test case, so you can now start searching for Name.
  - a) On the **Find/Replace Values** dialog box, select **Name** from the **Column** list box.
  - b) Click **Find Next** until SilkTest Classic highlights "Susan".  
Remember that you may have to change the direction of your search by clicking **Up** on the **Find/Replace Values** dialog box to locate this value.
  - c) Click **Replace** on the **Find/Replace Values** dialog box to replace that name with a link to the values in the data source.
5. Now search for and replace Zip codes in your script.
  - a) On the **Find/Replace Values** dialog box, select **Zip** from the **Column** list box.
  - b) Click **Find Next** until SilkTest Classic highlights "02421".  
Remember that you may have to change the direction of your search by clicking **Up** on the **Find/Replace Values** dialog box to locate this value.
  - c) Click **Replace** on the **Find/Replace Values** dialog box to replace that zip code with a link to the values in the data source.
6. When you have finished replacing values, click **Cancel** on the **Find/Replace Values** dialog box.

## Running Your Data-Driven Test Case

1. Click **Run Testcase** on the Data Driven workflow bar to run this test case.
2. Select **DD\_testcase1** and check **Use Sample Data from Script**.
3. Click **Run**.

Testcase1 should have run only once and the sample data from REC\_Products\_ grTest\_Products\_ been passed into it. Compare your file with the `gentest1_example.g.t` and `gentest1_example.res` found in the `<SilkTest installation directory>\examples` directory.

# Index

## D

- data-driven test case
  - creating 6, 8
- data-driven test cases
  - overview 4

prerequisites 5

## T

- test cases
  - data-driven 4