Your Guide to Negative & Positive Automated Tests

TestComplete

by SMARTBEAR

There are many types of tests from regression testing to destructive testing, from functional to non-functional. Though the type of testing will vary depending on our focus, we can all agree there is a need to test how our application reacts to valid or invalid user input. That is the pinnacle of negative and positive testing and understanding how to use these types of tests increases testing coverage of your application.

Automated Negative and Positive Tests in TestComplete

Contents

Positive and Negative Testing Overview

What is Positive Testing?

Positive tests are important but only cover a small amount of your applications behavior. If your positive tests pass, that is a great but you also need to see how your application reacts to erroneous user behavior. Positive testing is testing which attempts to show that a given module of an application does what it is supposed to do. A simple example, if you click on the Back button on a browser UI, the application should direct you to the previous page you were on; this would be a passing positive test. If you pressed the Back button and it refreshed the page; this would be a failed positive test.

What is Negative Testing?

Negative testing is an attempt to break the functionality of your application. For instance, say there was a text field in your application that should only accept digits as a valid data input the obvious negative test would be to enter letters into that field. The intended result would be that the user would receive an error explaining that this field does not accept letters and to try again. If the numerical field took letters as input data then this is a failed negative test.

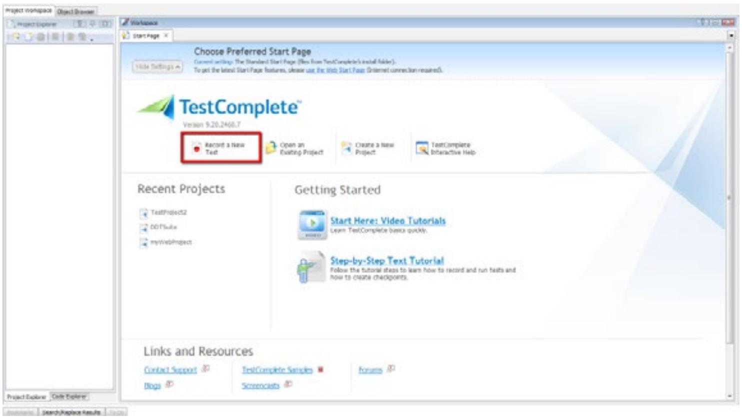
Using TestComplete for Negative and Positive Testing

Now that we have an understanding of positive and negative testing, how can we set up an automated test using these principles? In most cases we will be covering our automated negative and positive tests via a data-driven test. Not all negative and positive tests are data-driven tests but this is the best way to exemplify positive and negative tests in a testing automation environment.

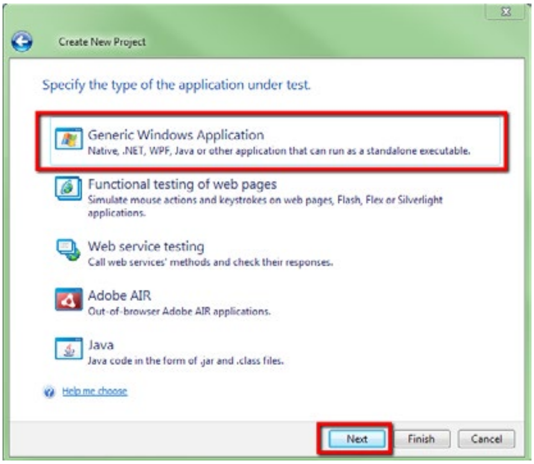
Creating a New Project

First, we will create a new TestComplete project. To do this:

1. If TestComplete is not already running, launch it now. Once it is running, select **Record a New Test** from TestComplete start page. Name your project and select a location to save it.



2. Select **Generic Windows Application**.



Specify the type of the application under test.

Generic Windows Application
Native, .NET, WPF, Java or other application that can run as a standalone executable.

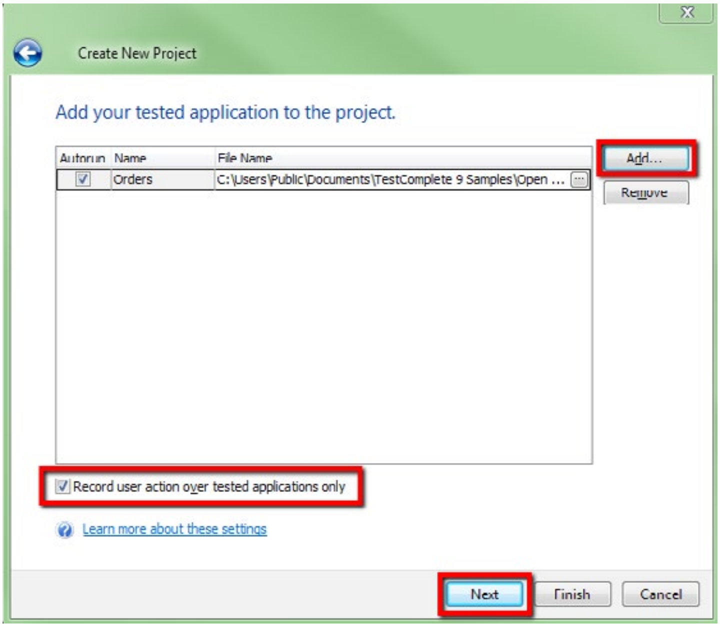
Functional testing of web pages
Simulate mouse actions and keystrokes on web pages, Flash, Flex or Silverlight applications.

Web service testing
Call web services' methods and check their responses.

Adobe AIR
Out-of-browser Adobe AIR applications.

Java
Java code in the form of .jar and .class files.

3. Click **Add** to add the application we are testing. For this example, we will test the Open Orders app provided with TestComplete. You usually find this application at **C:\Users\Public\Documents\Test-Complete 9 Samples\Open Applications\Delphi**. After you open this application, select **Record user action over test applications only**.



Add your tested application to the project.

Autorun	Name	File Name
✓	Orders	C:\Users\Public\Documents\TestComplete 9 Samples\Open ...

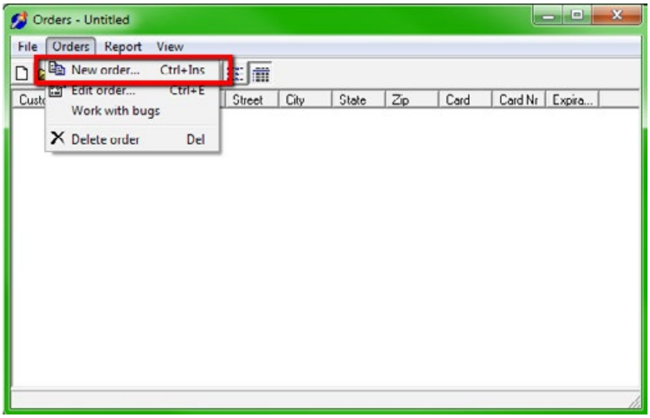
[✓] Record user action over tested applications only

4. Select **Next** to use the Test Visualizer during recording and play-back. This allows us to capture each action we make within a given test.

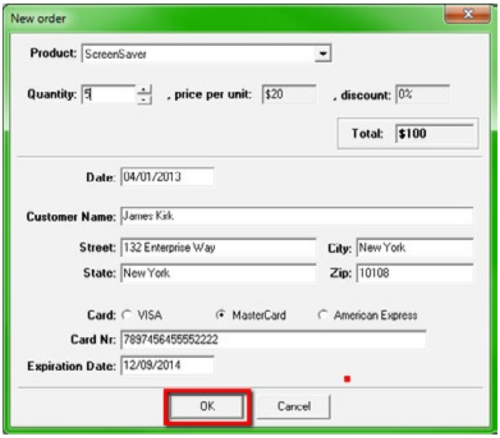
5. We will then select **Jscript** as out scripting language. You may use the language you prefer, but I am using Jscript for this example. Now your test recording will begin and the application we selected to test will automatically open.

Recording our test

1. Once the application is open let's start a test by entering data into this easy-to-use orders application. To start click on **Orders** and then select **New Order**.



2. In order to cover all of the data input fields we will enter all the required information for a single order. When done, click **OK**.


New order

Product: ScreenSaver

Quantity: 5 , price per unit: $20 , discount: 0%

Total: $100

Date: 04/01/2013

Customer Name: James Kirk

Street: 132 Enterprise Way City: New York

State: New York Zip: 10108

Card: VISA [✓ MasterCard] American Express

Card Nr: 7897456455552222

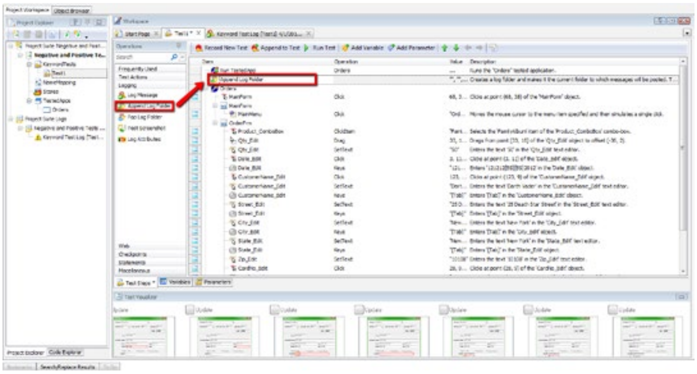
Expiration Date: 12/09/2014

3. The order will now appear in the orders database. Press **Stop** on the Test Recorder to finish the test.

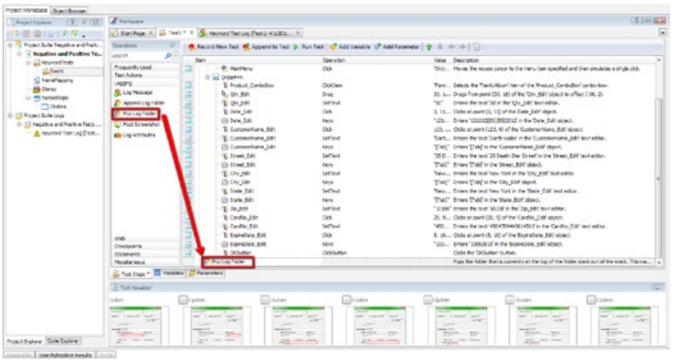
Setting up our automated test to increase test coverage

Now that we've recorded our automated test script, we will need to begin building our negative and positive tests via a data-driven test in order to quickly test all valid and invalid data for any given data field.

1. The first thing we should do is move the **Append Log Folder** to the test script at the point in which the Orders app opens. This will create a log folder where messages will be posted.



2. Next, move the **Pop Log Folder** operation to our test at the end after the click OKButton action. This activates the default folder that was active before the currently active folder. All of the messages that are posted to the log will be redirected to this "restored" folder.



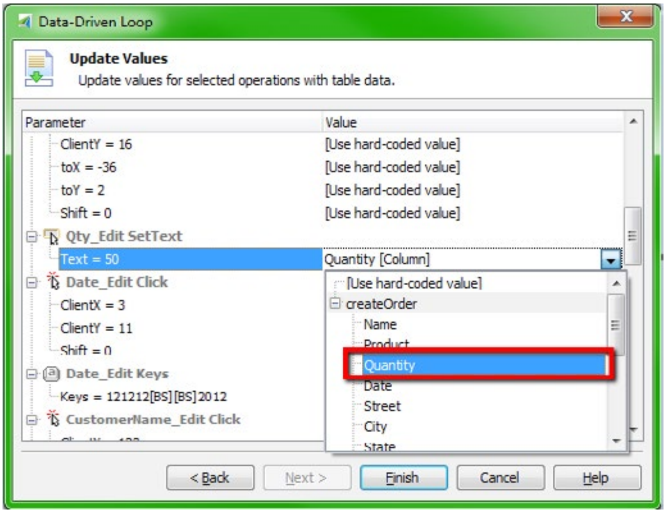
3. Now select everything from **Append Log Folder** to the **OKButton** action. Then right click and select Make Data Loop.

Name your new variable. For this example, let's name this createOrder because we are automatically creating orders with data from a spreadsheet. Now click **Next** and select **Excel worksheet**.

4. Select the file that will contain our data for the negative and positive tests. There is an example spreadsheet available in this folder: **C:\Users\Public\Documents\TestComplete 9 Samples\Data-Driven Testing**

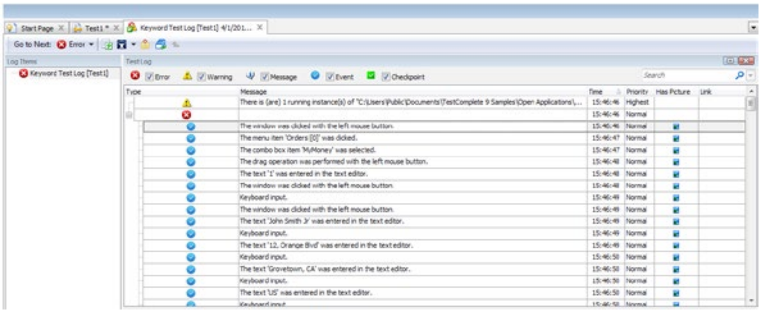
5. If your spreadsheet has the first row as column names, make sure to select **Treat the first row as column names**. Make sure your start record is **From the beginning** and the end record is **To the end**.

6. Now update your values to be whatever the column name is in the spreadsheet. For instance, for any given order in the spreadsheet, you want the quantity, name, etc. data to be entered into the correct field of the Orders app. When you are done click **Finish**.



Conclusion

We can now enter valid and invalid data into our spreadsheet and TestComplete can grab that data and send it through our application. Each data point will be logged and any errors will be represented at the end of the test (when all the data from your spreadsheet has been entered into the Orders application.)



Type	Message	Time	Priority	Has Picture	Link
⚠	There is (are) 1 running instance(s) of "C:\Users\Public\Documents\TestComplete 9 Samples\Open Applications\...	15:46:46	Highest		
✖		15:46:46	Normal		
●	The window was clicked with the left mouse button.	15:46:46	Normal	▣	
●	The menu item 'Orders|[0]' was clicked.	15:46:47	Normal	▣	
●	The combo box item 'MyMoney' was selected.	15:46:47	Normal	▣	
●	The drag operation was performed with the left mouse button.	15:46:48	Normal	▣	
●	The text '1' was entered in the text editor.	15:46:48	Normal	▣	
●	The window was clicked with the left mouse button.	15:46:48	Normal	▣	
●	Keyboard input.	15:46:49	Normal	▣	
●	The window was clicked with the left mouse button.	15:46:49	Normal	▣	
●	The text 'John Smith Jr' was entered in the text editor.	15:46:49	Normal	▣	
●	Keyboard input.	15:46:49	Normal	▣	
●	The text '12, Orange Blvd' was entered in the text editor.	15:46:49	Normal	▣	
●	Keyboard input.	15:46:50	Normal	▣	
●	The text 'Grovetown, CA' was entered in the text editor.	15:46:50	Normal	▣	
●	Keyboard input.	15:46:50	Normal	▣	
●	The text 'US' was entered in the text editor.	15:46:50	Normal	▣	
●	Keyboard input.	15:46:50	Normal	▣	

We just used an excel workbook in this example but keep in mind that you can use other inputs for your data, like MySQL or CSV file. Trying as many varieties of data as possible will increase your test coverage and further ensure the quality of your software.

BECOME A TESTING
SUPERHERO
We want you to be the best tester you can be!
Try TestComplete
30-Day Free Trial Evaluation

About SmartBear Software

More than one million developers, testers and operations professionals use SmartBear tools to ensure the quality and performance of their APIs, desktop, mobile, Web and cloud-based applications. SmartBear products are easy to use and deploy, are affordable and available for trial at the website. Learn more about the company's award-winning tools or join the active user community at http://www.smartbear.com, on Facebook or follow us on Twitter @smartbear and Google+.