



Technical Paper

Functional Testing of Adobe Flex Applications: Four Benefits of Using TestComplete

Adobe Flash applications with dynamic content present unique testing challenges, as they may contain not only the usual GUI elements like buttons, combo boxes and grids, but also animation and custom visual effects. TestComplete provides a broad set of capabilities for automated testing and verification on the Adobe Flash that benefit both developers and testers alike.

Introduction

Let's face it - software testing isn't an easy job. Depending on the size of the application and how established it is, your test team may have hundreds or thousands of tests to perform. These can range from simple usability checks to strict requirements coverage. Managing and executing these tests is an involved process.

Flash applications with dynamic content present some additional unique challenges, as they may contain not only the usual GUI elements like buttons, combo boxes and grids, but also animation and custom visual effects. The richness of the capabilities provided by the Flash Platform requires special consideration for thorough testing.

In this article, we will explore the benefits of testing a Flash application with TestComplete. TestComplete is an automated functional testing tool that fully supports the Adobe Flash application Platform, including Flex and AIR. While TestComplete has exceptional record and replay test automation, it also provides advanced testing capabilities which grant full access to an application's objects, methods and properties. TestComplete also has extensive data driven testing features, allowing tests to be easily linked to external data sets, and provides a comprehensive set of built-in verifications which allow validation of text, images, database contents and more. This feature set enables developers to thoroughly verify applications and solve even the most demanding testing challenges, such as testing custom GUI controls.

Binding Automated Tests to Data Sources

With the common record-playback approach to creating tests, what you get are tests that have the test data hard-coded into them. Even with the same general test logic, using a different data set for a test requires re-recording the test. Typically, you'll want to test your application with many different sets of input and expected values to verify that it properly handles various positive, negative and corner cases. Rather than recording a single instance of the test for each data combination, a much more efficient option is to make your recorded tests parameterized and run the same test using multiple data sets. This reduces both the time to create tests and maintain them over longer term.

Similarly to binding Flex controls to a data provider, TestComplete allows you to connect your automated tests to external data sources that can be used to drive the tests. Once an automated test is made data-driven, adding new test cases becomes as simple as adding a new row to the data source. TestComplete supports multiple data formats – CSV and Excel files, databases, and for a quick start TestComplete's internal data tables.

Creating data-driven tests with TestComplete is pretty easy. Once you have created the test steps representing the test logic, you can run them through TestComplete's Data-Driven Loop wizard (Make Data Loop in Figure 1):

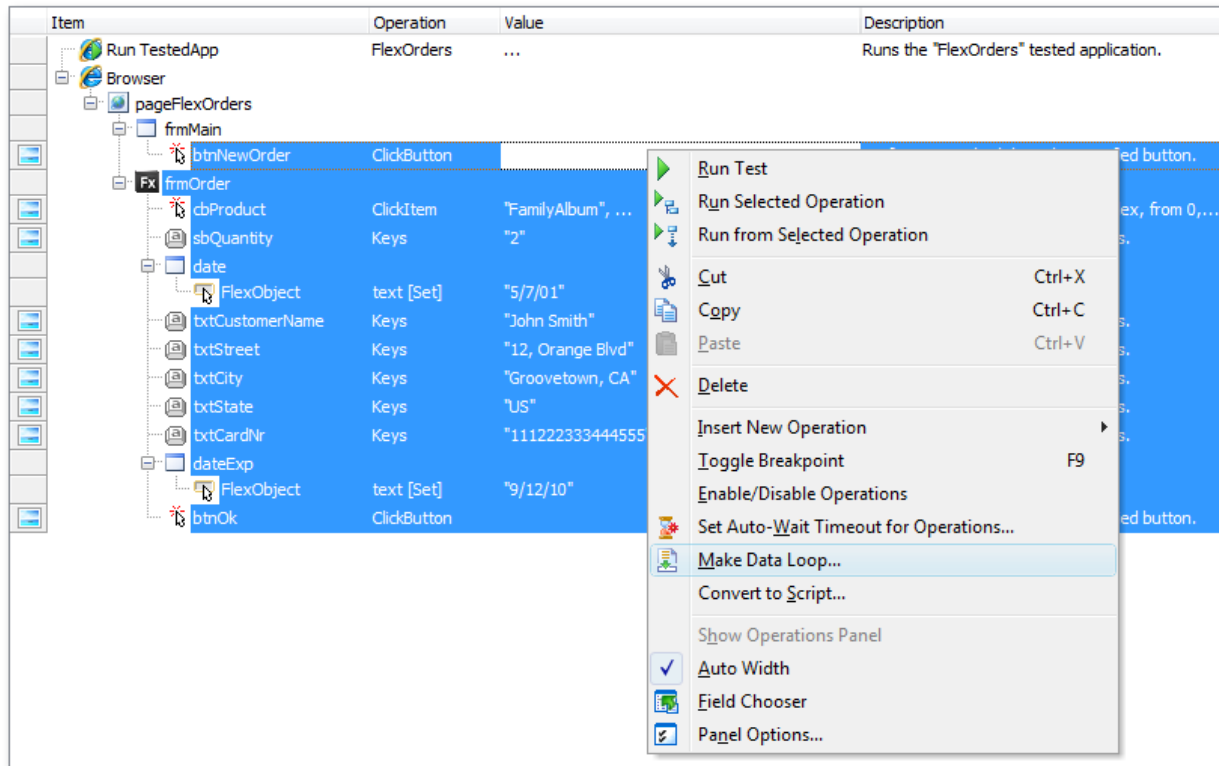


Figure 1. Parameterizing a sequence of test operations

The wizard analyzes/scans your test for constant values and opens the specified data source. You'll then be able to parameterize test steps by binding the variables with captured constant values to the data source fields.

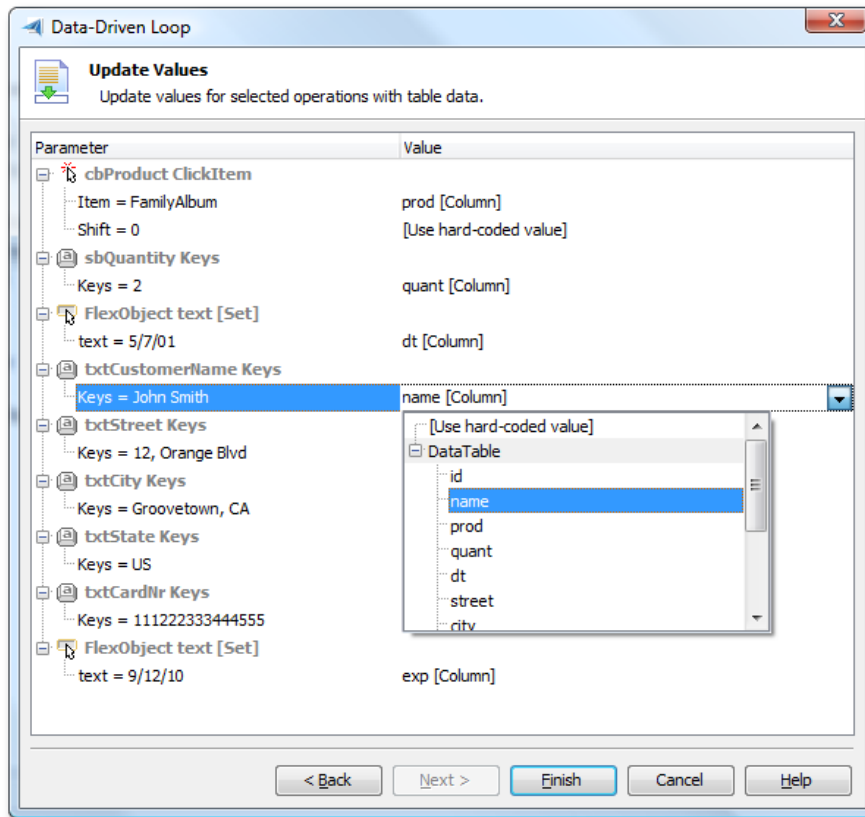


Figure 2. Replacing constant values with values from the data source

Even if you don't have a prepared data source, you can make use of TestComplete's internal smart data generator to automatically produce pseudo-random data for your automated tests. The data generator is seamlessly integrated into the Data-Driven Loop wizard, so that you can create a data source for your test on-the-fly while you parameterize it, or it can be launched at any time needed. With a wide range of data types available for you to choose from and combine – names, e-mail addresses, phone numbers, dates and much more – you can quickly and conveniently create large data tables with realistic data. The whole procedure takes just a couple of minutes – less than drinking a cup of coffee!

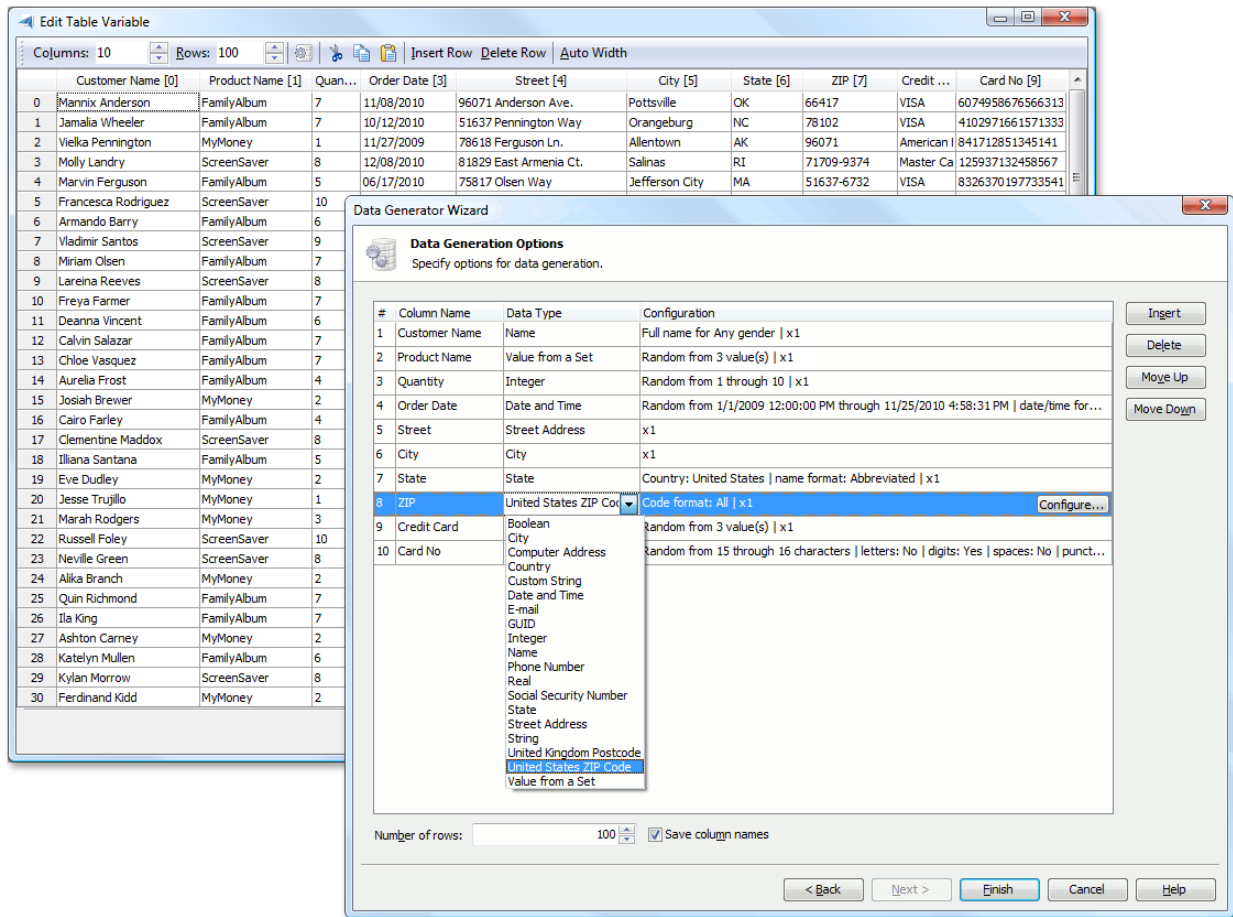


Figure 3. Generating a data source

Extending Tests Using Native ActionScript Objects and Properties

Sometimes developers extend the built-in Flex and Flash components with additional functionality required by the application, or even build components entirely from scratch. Therefore it is important that a test tool offers a reliable automation for custom controls as well. TestComplete effectively deals with customized components by providing testers with complete access to the application objects and their properties, including native ActionScript objects.

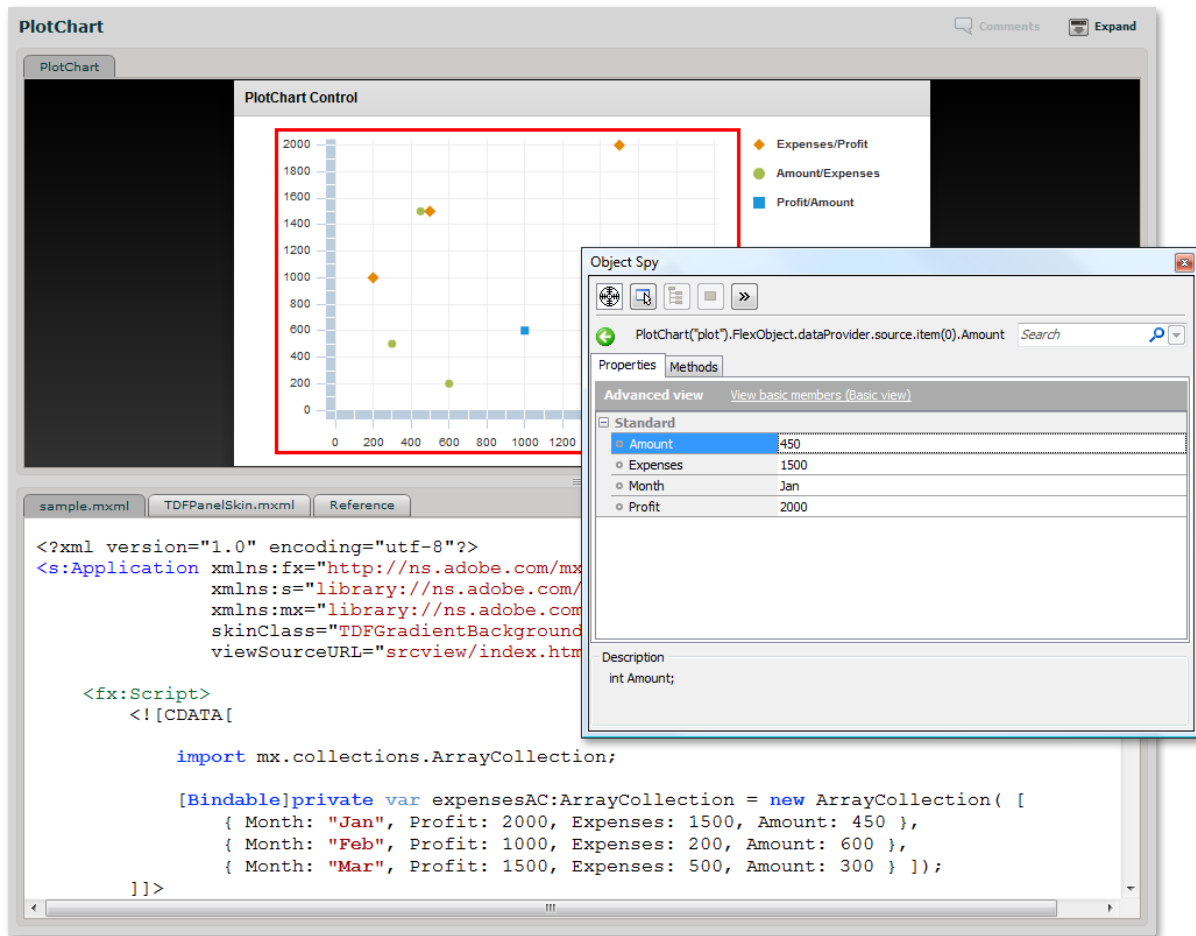


Figure 4. TestComplete's Object Spy displaying a chart's native dataProvider elements

How does access to native elements help automate testing of custom interface components?

First, this enables you to reliably change the state of a component programmatically from within the test instead of relying on coordinate-based clicks. From simple item selection to moving a slider to manipulating timelines in Gantt charts – anything is possible just the same way it would be done in the application's code.

Additionally, using native object properties allows you to perform verifications that otherwise would be difficult or ineffective to implement. For example, verifying a custom control's data is a lot easier by reading its underlying dataProvider rather than collecting data pieces from its elements.

Verifying Images and Graphics

When it comes to graphics and data visualization, Adobe Flash Platform equips developers with a vast array of options. Images, animations, charts – these are just a few examples of what's available to Flash and Flex developers for using in applications. While the graphic components enhance the application's visual appeal, they bring additional challenges for testers trying to test and verify the complex graphics objects.

How to verify that the localized logo changes according to the selected UI language? How to verify that the image zooms in or out properly? How to wait for an animation to complete before moving on? – These and other scenarios can be easily automated thanks to TestComplete’s image verification capabilities.

Simply point TestComplete to the graphic element you wish to verify, and you are presented with a range of options to customize the image verification process.

You can verify the entire image, as well as its specific portion defined by coordinates or a mask.

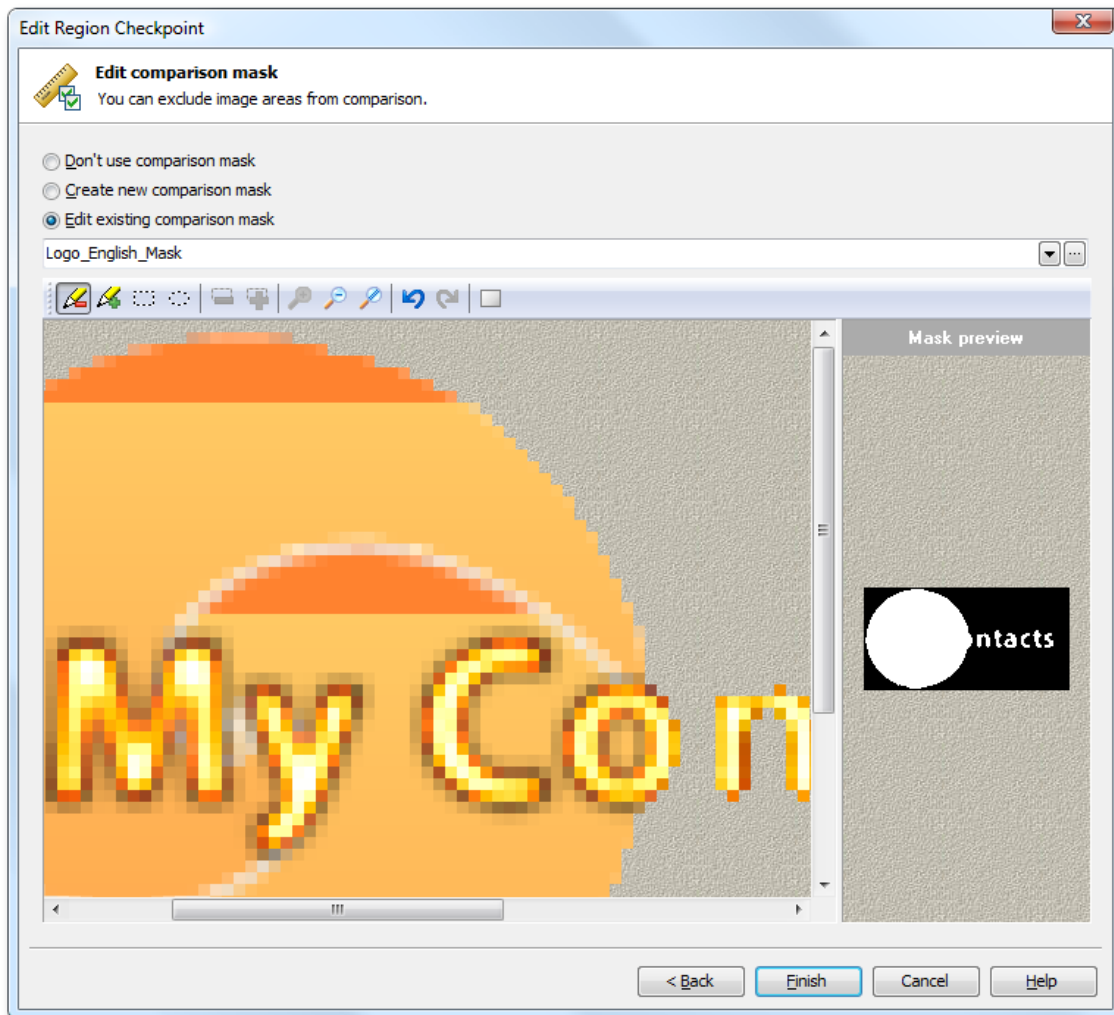


Figure 5. Defining mask for image verification

You can perform strict verification, which will fail even if a single pixel is different, or you can set a tolerance threshold for ignoring minor differences between the expected and actual images.

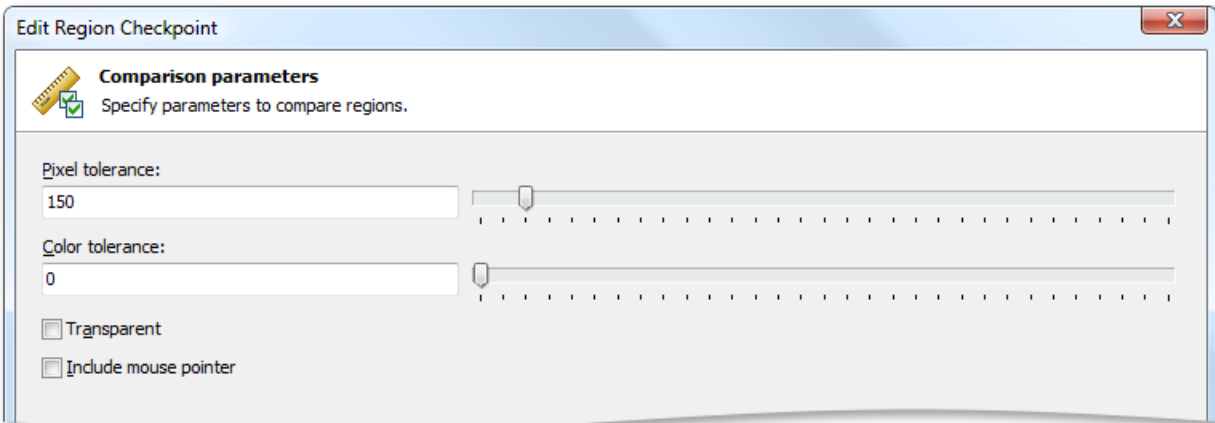


Figure 6. Setting comparison tolerance parameters

Last but not least, you can set a timeout for image verifications which will cause TestComplete to *wait for the expected image to appear*. This enables you to accurately synchronize the test execution with animations and visual effects in your application.

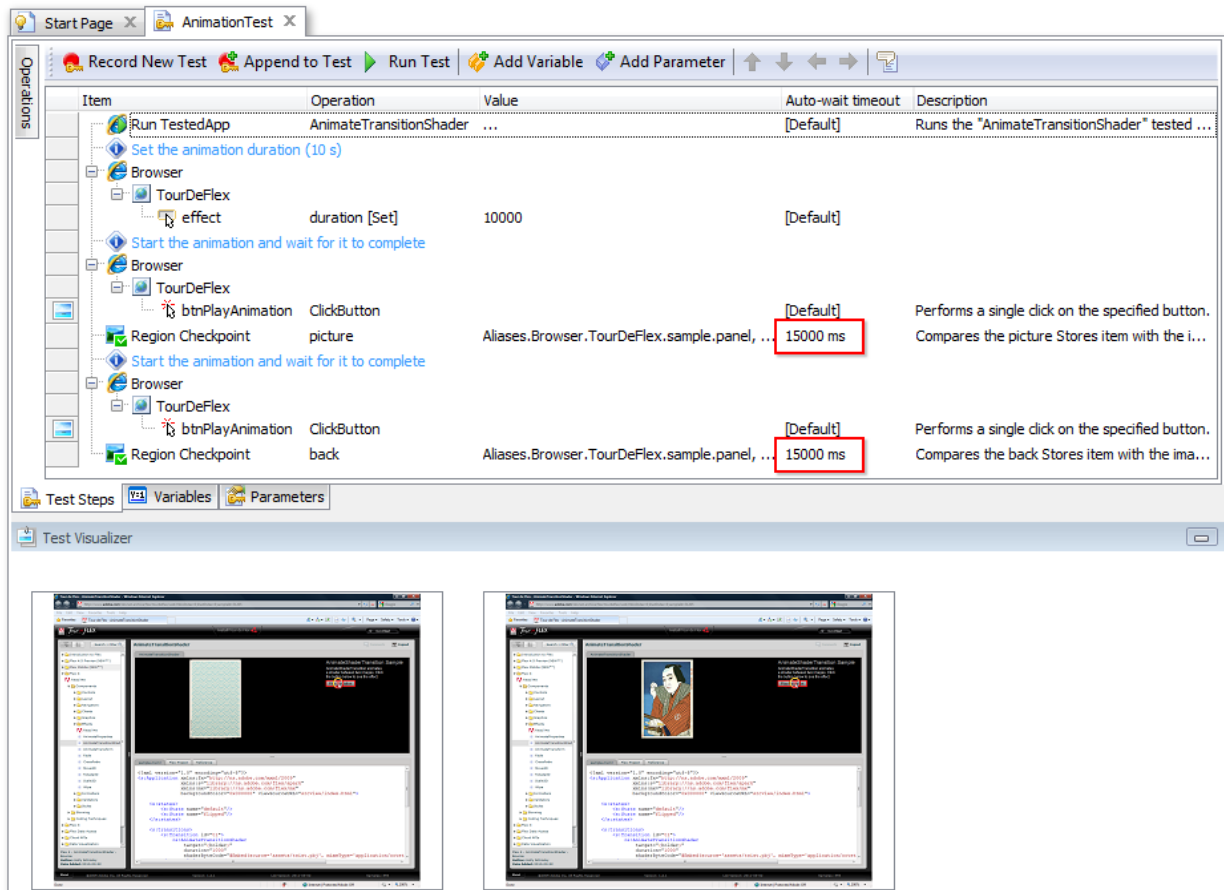


Figure 7. Image verifications with a timeout serve as “wait for image” operations

Update Tests Without Re-Recording

In most of the cases, recorded tests require modifications during their life span. The application itself may change, the tester may need to add more verification checkpoints, or new data needs to be fed through. Obviously, a much preferred option would be to change parts of the test or extend the available test with new elements, rather than recording the test from scratch.

One challenge with inserting new operations into the test is bringing your tested application to exactly the same state where you can capture the desired data or define the needed actions. In case of modern complex applications with dynamic content, this can be rather difficult and tedious.

TestComplete deals with this challenge by capturing detailed information about objects, properties and methods in your tested Flex application as you record or create automated tests. This enables you to modify and enhance your existing tests using the already captured information rather than re-recording the entire test. For example, you can insert new operations and checkpoints from the Test Visualizer screens by simply pointing to the desired object and then selecting the needed operation.

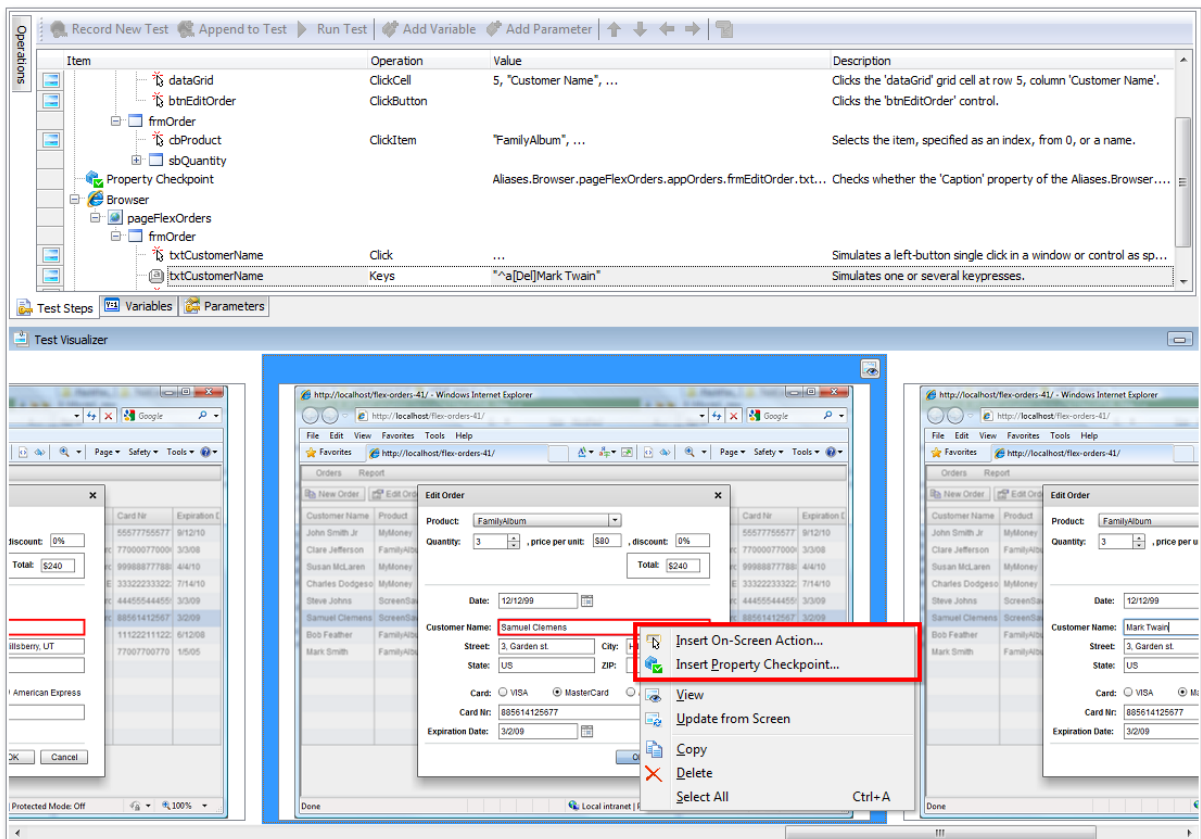


Figure 8. Inserting test operations from Test Visualizer

What's more important, TestComplete tests can be augmented in this way even when the application under test is not running. This gives you some extra productivity when editing and tweaking tests by reducing the context switching between TestComplete and other applications.

Where to Go From Here

This article discussed some advanced testing scenarios related to Flex and Flash test automation and showed how they can be solved with TestComplete. We invite you to [download TestComplete](#) and try it on your own Flex and Flash applications to automate the testing process. Share your Flex and Flash automation success stories with us, so others can be inspired!



For a live demo of TestComplete support for Flash, Flex and AIR applications, watch our webinar:



[Testing Adobe Flex and Flash applications with TestComplete 8: No magic required](#)

To learn more about automated testing with TestComplete, visit the following resources:

- > www.TestComplete.com
- > [TestComplete Webinars and Videos](#)
- > www.SmartBear.com

About SmartBear Software

SmartBear Software provides enterprise-class yet affordable tools that deliver quality and performance throughout the entire software quality lifecycle – enabling code quality; QA efficiency; and application performance anywhere – on the desktop, mobile and in the cloud. Our [collaboration](#), [code quality](#), [test automation](#), [performance monitoring](#) and [ALM](#) tools help more than 100,000 professionals to build and run some of the best software applications and websites in the world. Our users can be found around the world, in small businesses, Fortune 100 companies, and government agencies. For more information about SmartBear Software, our award-winning tools, or to join our active user community, visit www.smartbear.com, www.facebook.com/smartbear or follow us on Twitter [@smartbear](https://twitter.com/smartbear).

SmartBear Software

+ 1 978.236.7900

www.smartbear.com