

Product Review

Java GUI-Tests with qftestJUI

By Elke and Michael Niedermair

Besides unit tests, which test small parts of a system in isolation, tests that validate a system as a whole are becoming increasingly important. GUI based tests are able to detect various kinds or errors, not just those in the user interface itself. qftestJUI, a product from Quality First Software GmbH, is a tool that addresses itself to that task.

With qftestJUI, automating Java GUI tests is easy. It is a tool for the creation, execution and management of automated tests for Java/Swing applications with a graphical user interface. As of version 1.8, available since February 2005, Java applets can be tested in arbitrary browsers. Java WebStart and other start programs are easily integrated with this version. All JDK versions from 1.1 to 1.5 are supported.

Installation

As qftestJUI is itself written in Java, the software is platform independent. It is officially supported on Windows (XP, 2000, NT) and Linux. Many customers also use it productively on other platforms such as Solaris, AIX, HP-UX, Mac OS X. Installing qftestJUI with the installation wizard on Linux did not cause any problems and was straightforward.

First steps

First time users of qftestJUI get valuable support from the vendor. The product comes with a 450 page manual which includes a user's guide and a reference manual. For quick introduction, a tutorial (128 pages) is also included. It provides easy to understand test scenarios and can be worked through in about two hours. Furthermore, the software is fully localized in Eng-

lish and German. If any problems remain, the FAQ on Quality First Software's web site (www.qfs.de) or the mailing list (qftestJUI-list@qfs.de) with the associated mailing list archive can be turned to for help.

Features of qftestJUI

Thanks to its intuitive user interface, qftestJUI is also well suited for users with little or no programming experience. Even complex tests can be created in short time with little effort. The product is flexible and adapts to changes in the system under test, has functionalities for recording and filtering and even supports tables and trees with dynamic content. Replay of recorded tests is facilitated by unlimited access to all recorded sequences. The product makes use of a sophisticated algorithm for component recognition so that the maintenance effort can be minimized.

qftestJUI provides of a number of modern control structures like procedures and variables, loops and conditions and text processing with regular expressions. With the integrated scripting language Jython, the tool gives access to the complete Java API of the application under tests. It produces XML/HTML-based test reports including screenshots.

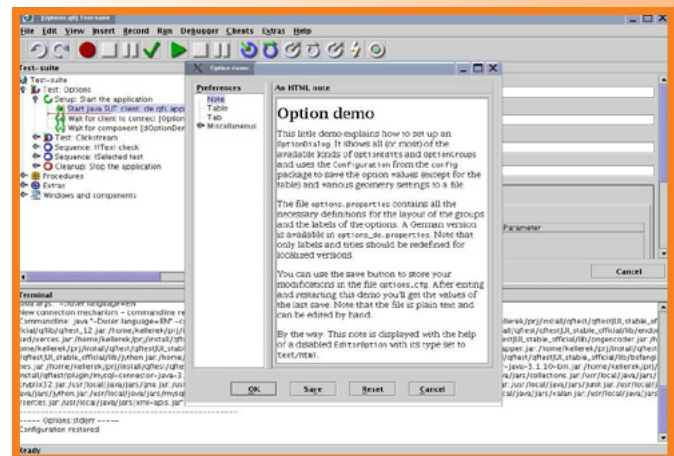


Fig. 2: Screenshot of the main window of qftestJUI with demo application

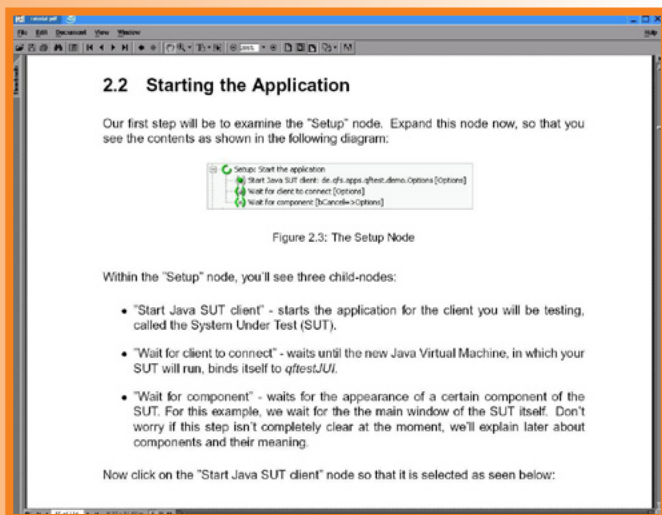


Fig. 1: Screenshot tutorial

Functionality

With qftestJUI multiple programs can be run simultaneously so that complex systems can be tested as a whole.

When the system under test (SUT) is started from qftestJUI, Java applications are executed in separate virtual machines (JVM). In doing so, special code is introduced that first sets up an RMI connection to qftestJUI and installs its own EventQueue before passing control to the SUT. The EventQueue is necessary to keep track of the components of the SUT and for recording events. Special provisions are made to prevent conflicts in case the SUT also installs its own EventQueue.

When a test sequence is replayed by qftestJUI, the event data is sent together with information about the target component to the SUT via RMI. There, a flexible algorithm determines the component to which the event pertains. It does not rely on the geometry of the component alone, but also takes the hierarchical structure of the GUI into account, along with miscellaneous component information like titles or labels.



Finally the event is reconstructed and inserted artificially into the system's EventQueue. For the SUT there is no difference between a real user event and an event coming from qftestJUI.

What qftestJUI cannot do yet

Unfortunately, other Java technologies like J2EE, Windows and Web based applications and Eclipse/SWT are not (yet) supported. Also, qftestJUI doesn't come with its own test management system, but interfaces to test management tools from other vendors are available. Code coverage analysis also is possible only with suitable tools. So the producer still has many options for future versions.

qftestJUI version 2, the first major upgrade, is planned for late this/early next year. Important topics are support for SWT and further improved usability. Many details will be honed. Furthermore there are plans for some test management extensions

(planning, administration, result analysis, reporting), including an interface to the open source test framework FIT (<http://fit.c2.com>), which enjoys increasing popularity. It will also be possible to integrate unit tests so they can be called from qftestJUI, with all the advantages of unified test management and reporting.

Load tests, which are already partly available, are another very interesting extension. qftestJUI can drive multiple test clients (from multiple threads) concurrently on one desktop and control them individually.

Conclusion

qftestJUI has shown that it can test Java Swing applications very well. Its ease of use and the great support (manual, tutorial, FAQ, mailing list) make it easy to learn how to use qftestJUI and quickly introduce it into a productive environment.

Product Information at a Glance

Produced by	Quality First Software GmbH, 82538 Geretsried, Germany www.qfs.de
Current version	1.08.2
Lizenzpreise (incl. VAT)	1 license: EUR 1295.- 2 - 5 licenses: EUR 1205.- each 6 - 10 licenses: EUR 1125.- each more than 10 licenses: Individual offer Academic licenses are available on request. For open source projects, a freely extensible evaluation license is offered.
Supported operating systems	Windows (XP, 2000, NT) Linux, Solaris, AIX, HP-UX, Mac OS-X
Supported JDK versions	JDK 1.1 through 1.5