



QF-TEST

Professional GUI Testing for Java and Web

QF-TEST Evaluation Report

Evaluation of GUI Test Tools for Java Swing Applications

by Klaus Berg, Senior Engineer,
Siemens Corporate Technology, Munich, Germany
July 2009

GUI tests are a vital step in the software testing life-cycle because they are performed from the view of the end user of an application. Test automation of this phase is the final goal as it is already done with unit or integration tests.

When you search the Internet for such a task you will find a broad spectrum of functional test tools that are either „general purpose“ test tools or dedicated to a specific kind of GUI. My job was to find a suitable test tool that is able to perform automated tests for a complicated Java Swing GUI (provided as a Java Web Start application).

Of course, you can start evaluating all the „big players“ in this area but „big player“ also means „big money“ in terms of license fee and maintenance cost. The other possibility is to look for vendors that are specialized in Java Swing GUI testing if this is all you want. Such a first preselection was my starting point and QF-Test was one of my first candidates for a deeper analysis.

Open source tools may be a choice too for every user that is looking for a low cost solution. However, in my opinion these tools are capable of handling simple GUI tests but reach their limits with complex GUIs (custom components, etc.) if you do not invest time and money to do a lot of „customization“ (it's open source, so you are free to extend them). But you can also decide to take the middle road and then you should definitely give QF-Test (for Java) a try.

For a tool comparison I have created more than 30 evaluation criteria (annotated with different priorities) like

- record and playback possibility
- smart GUI component recognition
- object mapping
- image checks
- test failure and error recovery
- support for data-driven testing
- use of popular scripting languages
- base ground for keyword-driven testing (like QF-Test's parameterized procedures)
- integration into popular build and/or continuous integration systems (like Ant/Maven or CruiseControl)
- test execution/result reporting using XML and HTML
- multi-screen support
- drag&drop support
- possibility to recognize complex custom components
- license & maintenance costs
- quality of support (availability of skilled resources, validity of responses from the help desk)
- online resources (like active forums rather than ringing the support line)
- speed of responses
- quality of online documentation
- ...

Having these evaluation criteria as well as dedicated test scenarios for the real system under test I have to say that QF-Test succeeded in fulfilling all my requirements with a score of more than 90%.

That result placed QF-Test above all other evaluated commercial and open source tools -- at least for my specific requirements and target. **However, I think you will hardly find another Java Swing GUI test tool that gives you a better value for your money.**