



QF-Test/swt: Kommerzielles Werkzeug für automatisierte SWT-Tests

Automatisierte Kontroll-Untersuchung für RCP

>> MARKUS STÄUBLE

Mit jeder neuen Version von Eclipse werden der verwendeten GUI-Bibliothek SWT auch wieder neue Funktionen verpasst. Diese Möglichkeiten möchte der Entwickler von Plug-ins oder RCP-Anwendungen dem Anwender natürlich nicht vorenthalten. Der Anwender hat trotz der wunderbaren neuen Möglichkeiten selbstverständlich den Anspruch, getestete Software zu bekommen. Solche Tests fangen bei Unit-Tests an, dürfen hier aber auf keinen Fall aufhören. Auch die Oberfläche gehört getestet. Der Anspruch des Entwicklers geht hier auf Automatisierung. Geht das auch bei RCP?

O bwohl sprichwörtlich der erste Eindruck täuscht, lässt man sich von diesem ersten Eindruck doch sehr oft in seiner Meinung prägen. Übertragen auf Software ist dieser erste Eindruck zumeist der Kontakt mit der Benutzeroberfläche. Selbst wenn dahinter eine bis ins Detail ausgeklügelte Logik steckt, Fehler in der Oberfläche werden oft nicht verziehen und können bei einem schnellen Test schon das K.o.-Kriterium sein. Um nicht den Ruf von Bananensoftware – Software reift beim Kunden – zu bekommen, ist es zwingend, für professionelle Software auch die Oberfläche zu testen. Bei einem manuellen Ansatz kann dieses Testen schnell sehr langwierig werden und Dinge können vergessen werden.

Darum ist auch hier der Anspruch an Automatisierung berechtigt. Viele

Anwender bedienen sich für Desktopanwendungen bei dem reichhaltigen Angebot der Eclipse Rich Client Platform (RCP) [1]. Die Oberfläche von Eclipse RCP basiert bekanntlich auf SWT. Ein Werkzeug, um solche Oberflächen automatisiert zu testen, ist QF-Test/swt der Firma QFS [2].

Was ist im Angebot?

Von QF-Test gibt es mehrere Versionen. Zunächst QF-Test/swing zum Testen von Swing-Anwendungen. Dann QF-Test/swt zum Testen von SWT-Anwendungen. Und letztlich QF-Test (oder QF-Test/suite), das beide Funktionen vereint. Für unsere Intention reicht QF-Test/swt völlig aus. Bei QF-Test/swt handelt es sich um ein kommerzielles Produkt der Firma QFS. Dieses grafische Tool lebt sehr viel vom Capture-

Replay-Prinzip. Das heißt, möglichst vieles Aufzeichnen und diese Dinge dann durch weitere punktuelle Tests anreichern. Dadurch gelingt es auch relativ schnell, zu Ergebnissen zu kommen. Die Tests selbst können über Prozeduren und Auslagerung in andere Dateien sehr modular aufgebaut werden. Es gibt das Prinzip von Vorbereitung und Aufräumen, ähnlich dem z.B. aus JUnit [3] bekannten Prinzip von *setup* und *tearDown*.

Es steht ein Debugger zur Verfügung und auch eine ausführliche Protokolldatei, die im mitgelieferten Viewer gut gelesen werden kann. Das Protokoll hat dabei den gleichen Aufbau wie Test-Suites und somit kann ein Fehler sehr gut zugeordnet werden. Für den Ausbau der aufgezeichneten Elemente steht die Skriptsprache Jython [4] zur Verfügung. Diese Sprache ist dem Java-Entwickler sehr nahe und daraus kann auch auf die bekannten Java-Elemente zugegriffen werden. Preislich [5] liegt eine Einzellizenz (Stand Oktober 2008) bei 1595 Euro. Bei einer erhöhten Lizenzanzahl staffeln sich die Preise. Zusätzlich zum Produkt wird ein Pflegevertrag angeboten. Dieser ist sowohl für Updates als auch für Support notwendig.

Ein kleiner Testlauf am großen Vorbild

Für eine Evaluierung kann QF-Test/swt einfach von der Webseite [6] heruntergeladen werden. Die heruntergeladene

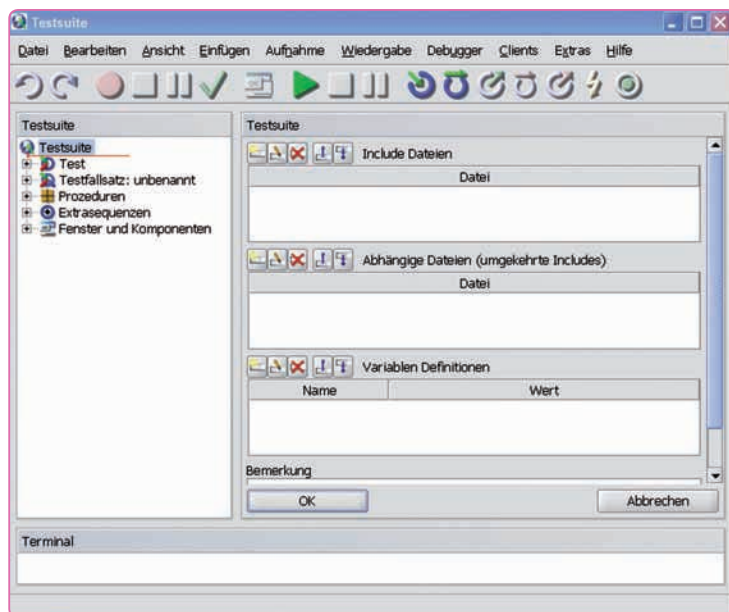


Abb. 1: Das Arbeitsfenster von QF-Test/swt nach dem Start

Der einfachste Weg, um Eclipse zu starten, ist über den mitgelieferten Starter (unter Windows: *eclipse.exe*). Dieser soll auch für die Tests verwendet werden. Somit ist die erste Option (ausführbares Programm) innerhalb des Dialogs auszuwählen. Nun muss die Startdatei *eclipse.exe* ausgewählt werden – optional kann ein Arbeitsverzeichnis für die Anwendung gesetzt werden, dies ist bereits vorgelegt und muss für unseren Test nicht geändert werden. Nun folgt der wichtigste Schritt im Prozess, nämlich die SWT-Instrumentierung (Kasten: „Instrumentierung“). Die Option *Prozeduraufruf SWT-Instrumentierung generieren* (Abb. 3) muss aktiviert werden.

Nun geht es noch darum, dem SUT einen Namen zu geben. Dies ist notwendig, um den Testkandidaten auch eindeutig identifizieren zu können. Im gegebenen Fall bietet sich „eclipse“ für einen Namen an. Der letzte Schritt ist der Abschluss des Wizards (es besteht die Möglichkeit anzugeben, ob das SUT gleich gestartet werden soll oder nicht). Nach Abschluss des Wizards wurde nun ein Knoten *Vorbereitung* erzeugt. Mit diesem kann das SUT gestartet werden (Abb. 4).

Nun können Sie einfach über den Recorder Abläufe aufnehmen (roter Knopf in der Toolbar), diese wiederspielen und mit Tests anreichern. Diese Tests können bis zu einem gewissen Grad auch direkt (grüner Haken) aufgenommen werden (Abb. 5).

Wer hilft, wenn es klemmt

Die ersten Schritte mit QF-Test werden sicherlich einfach von der Hand gehen. Um auch ausgefeilte Test-Suites erstellen zu können, stehen mehrere Hilfen zur Verfügung. Einmal wird ein Handbuch (HTML und PDF) mitgeliefert. Darüber wird ebenfalls ein empfehlenswertes Tutorial mitgeliefert, das sehr zur Einarbeitung zu empfehlen ist. Neben diesen beiden Quellen gibt es eine aktive Mailinglist [8]. Falls dies nicht mehr hilft, steht ein kostenpflichtiger Support zur Verfügung [9].

Ausblick auf Morgen

Zuletzt soll ein kleiner Ausblick auf die kommende neue große Version 3 von QF-Test gegeben werden. Einige Highlights (komplette Liste aller Features unter [10]) der Version 3 sind nachfolgend aufgeführt:



Abb. 2: Unterschiedliche Version von instrumentierten SWT-Plug-ins

Windows gestaltet sich einfach in dem Ausführen des EXE-Installers (Größe: ca. 64,8 MB). Nachdem nun die angebotenen Möglichkeiten grob auf dem Tisch sind, soll ein erster Testlauf von QF-Test/swt ein erstes Gefühl für die Art und Weise, wie damit getestet werden kann, geben. Wenn wir über SWT sprechen, was liegt dann näher, als den großen Vorreiter in der Verwendung dieser Bibliothek zu nehmen – Eclipse. Für diesen Test wurde die aktuelle Version 2.2.4 von QF-Test/swt verwendet. Nach der Installation kann das Tool einfach per Doppelklick gestartet werden. Nach dem Start präsentiert sich das Arbeitsfenster und ist bereit für die Arbeit (Abb. 1).

Um eine Anwendung mit QF-Test/swt testen zu können, muss diese auch daraus gestartet werden. Die Begrifflichkeit für die zu testende Anwendung innerhalb des Tools ist SUT (System unter Test). Der richtige Start der Anwendung ist ein essenzieller Bestandteil des Testens, darum wird zur Erstellung der Startprozedur auch ein Wizard angeboten. Dieser ist über *EXTRAS | SCHNELLSTART | WIZARD* erreichbar. Die erste Frage, die gestellt wird, ist nach dem Typ der Anwendung. Folgende Typen von Anwendungen stehen zur Auswahl:

- Startskript (z.B. *batch*-Datei) oder auch ein ausführbares Programm
- Java-WebStart-Anwendung
- Applet
- *jar*-Archiv
- Java-Klasse



Abb. 3: Vergessen Sie nicht die SWT-Instrumentierung im Schnellstart-Wizard

Version bietet alle Funktionen wie die Vollfunktion. Ohne Lizenz können aber lediglich die mitgelieferten Test-Suites geladen werden bzw. erstellte Test-Suites können nicht abgespeichert werden. Es besteht aber die Möglichkeit, eine 30-tägige Testlizenz anzufordern, die das Speichern und Öffnen anderer Test-Suites ermöglicht. Die Installation unter

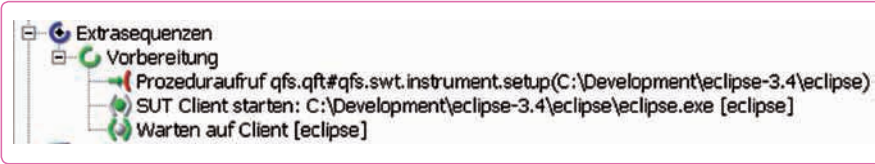


Abb. 4: Erzeugter Knoten Vorbereitung zum Start von Eclipse als SUT



Abb. 5: Die Toolbar von QF-Test/swt

- Anbindung an HP/Mercury Quality Center und Imbus TestBench
- Groovy als zweite Skriptsprache
- Verbessertes Jython- und Groovy-Skripteditor mit Syntaxhervorhebung
- Geteilte Protokolle ermöglichen lang laufende Tests ohne Speicherprobleme

Im „großen“ QF-Test kommt zusätzlich das Webmodul (QF-Test/web) zum automatischen Testen von Webanwendungen hinzu. Erwähnenswert ist sicherlich, dass viele neue Funktionen in der aktuellen Betaversion 2.9.2, die heruntergeladen werden können, bereits heute angesehen werden können. In Abbildung 6 ist z.B. der neue Skripteditor mit Syntaxhervorhebung abgebildet.

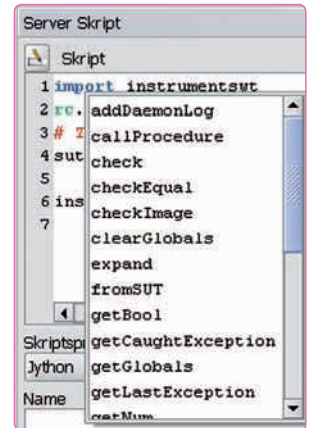
Lassen Sie Ihre Anwendung entscheiden

„Ist es das richtige für mich?“ Das werden Sie sich vielleicht jetzt fragen. Die erste gedankliche Runde für diese Frage

ist nach diesem Artikel sicherlich schon gedreht. Ob das Tool die groben Anforderungen abdeckt, ist anhand der Webseite einfach herauszufinden. Alle weiteren Fragen sollten Sie Ihrer Anwendung überlassen. Wie bei jeder anderen Toolentscheidung ist auch hier wärmstens zu empfehlen, das Angebot der Evaluierung an der eigenen Anwendung anzunehmen. Eine Testlizenz für die Dauer von 30 Tagen, die es auch ermöglicht, Test-Suites abzuspeichern, kann über die Webseite von QFS angefordert werden. Auch Support durch den Hersteller kann in dieser Zeit in Anspruch genommen werden. Nach den ersten Runden der Aufzeichnung kommen bald die Stellen, an denen die Tests komplexer werden und Sie mit Jython oder mit Groovy (in der Version 2.9.2 enthalten) in Berührung kommen. Hier wird sich zeigen, ob QF-Test/swt für Ihre Anwendung und Ihre Infrastruktur passend ist oder eben nicht.

Insgesamt handelt es sich bei QF-Test/swt um ein intuitives Tool, welches

Abb. 6: Der neue Skripteditor bietet Syntaxhervorhebung und Quellcodevervollständigung



sicherlich eine Chance verdient hat. Empfehlenswert ist auch, die 30-Tage-Evaluierungszeit auszunutzen, denn es gibt viele Dinge zu entdecken, die in diesem Bericht nur angekratzt bzw. gar nicht gezeigt (z.B. Testen mit mehreren SUTs) werden können. Auch die Preisgestaltung des Produkts ist nicht unattraktiv. Bei einem Einstieg lohnt es sich, das Tutorial zu durchlaufen, um etwas mit der Arbeitsweise vertraut zu werden. Lassen Sie es auf einen Test ankommen!

>> Links & Literatur

- [1] Eclipse RCP: wiki.eclipse.org/index.php/Rich_Client_Platform
- [2] QFS - Quality First Software: www.qfs.de/de/index.html
- [3] JUnit: www.junit.org
- [4] The Jython Project: www.jython.org/Project/
- [5] QF-Test, Lizenz- und Supportpreise: www.qfs.de/de/qftest/pricesswt.html
- [6] QF-Test, Download und Installation: www.qfs.de/de/qftest/download.html
- [7] Java Accessibility: java.sun.com/j2se/1.4.2/docs/guide/access/
- [8] QF-Test Mailinglist: www.qfs.de/maillman/listinfo/qftest-list
- [9] QF-Test, Produktsupport: www.qfs.de/de/qftest/support.html
- [10] QF-Test, Version 3 Beta: www.qfs.de/de/qftest/30beta.html

Instrumentierung - eine Änderung muss sein

Beim allerersten Start wird gefragt, ob die installierten JREs instrumentiert werden sollen. Die Instrumentierung ist notwendig, damit sich QF-Test/swt transparent zwischen JRE und Anwendung hängen kann. Somit sollten Sie die automatische Instrumentierung wählen. Dies hat keine negativen Auswirkungen für den Start ohne QF-Test/swt und kann im Bedarfsfall ohne Probleme rückgängig gemacht werden. Im Falle einer JRE-Instrumentierung bedient sich QF-Test/swt dem offiziellen Accessibility-Interface [7]. Für die Änderung werden nachfolgende Änderungen in die Datei *accessibility.properties* (*lib*-Verzeichnis der jeweiligen JRE) eingetragen:

```
QF-Test.connector.logfile=C:\Program Files\qfs/QF-Test/log/qfconnect.log
```

```
QF-Test.versionhome=C:\Program Files\qfs/QF-Test
assistive_technologies=de.qfs.apps.QF-Test.start.
Connector
```

Für SWT hilft dieses Interface nicht weiter und deshalb ist für SWT-Anwendungen noch ein weiterer Schritt notwendig. Die SWT-Instrumentierung tauscht nämlich die jeweilige SWT-Bibliothek aus. Speziell für die unterschiedlichen Eclipse-Versionen wird jeweils ein spezielles Plug-in mit ausgeliefert (Abb. 2).

Bevor Sie mit der Instrumentierung von SWT-Anwendungen beginnen, sollten Sie die ursprüngliche SWT-Bibliothek bzw. im Falle von Eclipse das Plug-in *org.eclipse.swt.win32.win32.x86_VERSION.jar* an eine andere Stelle kopieren, um hier auch zum Ausgangspunkt zurückkommen zu können.



Markus Stäuble ist CTO (Head of Technology) bei namics (deutschland) GmbH, einem führenden IT- und Webdienstleister. Darüber hinaus ist er freier Autor von Fachartikeln.