Automated Qt GUI Testing with Squish

Functional GUI testing is an essential part of development and QA when creating sophisticated modern GUI applications. Manual testing alone cannot review an application for regressions within an acceptable time frame. Automated GUI tests quickly produce reliable and reproducible results, and execute in a fraction of the time. Automated GUI testing enables test engineers to expand testing coverage and rapidly find regressions so fixes can be make early, saving both time and money.

The creation of a maintainable and time-proven automated GUI testing framework requires an automation tool that embodies a deep knowledge of the underlying GUI technologies being tested.

froglogic's popular cross-platform, multi-technology GUI testing tool, Squish, supports automating testing for all kinds of modern GUI applications, with dedicated and comprehensive support for:

- Qt, QML, QtQuick and QtWebKit
- Java GUIs
- Native Windows Controls
- Mac OS X Cocoa and Carbon
- iOS Native and Web GUIs
- Android Native and Web GUIs
- Web and Flex in multiple browsers
- ... and more...

Squish has many other features, including:

- ☑ Powerful and easy to use test development environment
- ☑ Record and replay with powerful scripting capabilities
- ☑ Choice of non-proprietary scripting language
- ☑ Access to all the tested application's objects
- ☑ Interactive Object Spy & Troubleshooting Utility
- ☑ Command line toolset and 3rd-party integrations for unattended test automation
- ☑ Open data formats (e.g., plain text configuration files and plain text and XML/HTML reports)
- ☑ Support for hybrid applications (Qt+Web, Qt+ActiveX, Java+Windows, iOS+Web and more)
- ☑ Source code availability

Licensing
Named User Licenses and Group Licenses with concurrent seats are available. All licenses are perpetual and include a free 12-month Support & Updates Subscription.
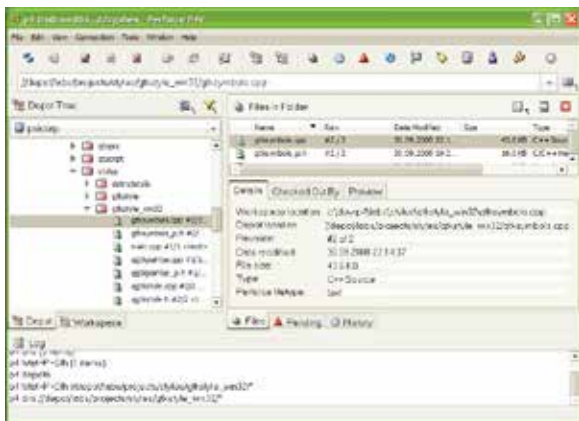
More Information
Please visit www.froglogic.com or contact us at sales@froglogic.com for additional questions or a live demonstration.

Squish for Qt: Success at Perforce

Perforce Software, founded in 1995, markets and supports Perforce, the Fast Software Configuration Management (SCM) system.

Recently, Perforce adopted Squish for GUI performance testing of their SCM system's flagship GUI, the Perforce Visual Client (P4V). We discussed Perforce's use of Squish with Tim Brazil, a Perforce Performance Lab engineer.

Why Squish?

Tim chose Squish to round out his testing arsenal and help ensure that performance remained high as new features and bugfixes were applied to P4V. P4V, written in C++ and using the Qt GUI library, provides a graphical interface on Windows, Mac OS X, Linux, Solaris and FreeBSD.

Squish provides a complete GUI-based testing IDE capable of recording and playing back GUI tests, but for Tim, it was also Squish's command line support that was particularly appealing.

"I prefer to work in a command-line environment that facilitates the use of scripting languages," Tim explained. "It is evident that froglogic's approach took engineers like myself into account when they designed the product. For example, features like the envars file, suite.conf file, squishserver, and squishrunner, allow me to design a fairly complex test environment with relatively little work."
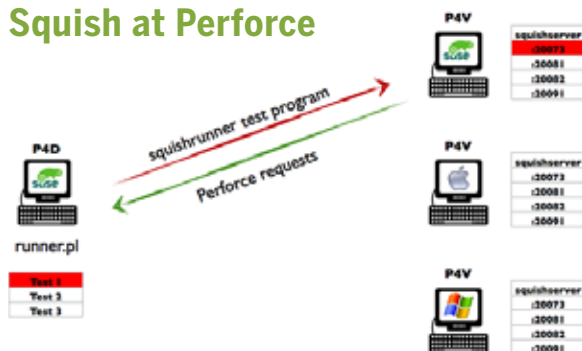
> "It is evident that froglogic's approach took engineers like myself into account when they designed the product."

Tim has many years experience in the software testing industry and is a strong proponent of using both automated and manual testing.

"The benefits of automated testing include reliability, repeatability, comprehensiveness, and speed, Furthermore, automated tests can quickly and dependably navigate through test scenarios and are, therefore, more likely to uncover subtle timing problems." He pointed out that manual testing alone is often both slow and fallible, and that even with a comprehensive test plan at hand it can be difficult to reliably recreate the exact test actions and environment.

Automated GUI Testing with Squish

Squish at Perforce

Tim uses Squish to test the performance of P4V, specifically on Windows Professional, Mac OS X, and SLED (SUSE Linux Enterprise Desktop). Squish is used to test the nightly build as well as the previous three P4V releases. A daily performance report is generated, identifying performance trends and pinpointing any possible areas of concern as P4V continually evolves.

Squish in Practice

"Squish's objects.map is unique and greatly facilitates test readability, robustness, and maintenance."

We asked Tim what features of Squish he liked most, apart from the multiple scripting language and command-line tools support he'd already mentioned. Squish's Object Map came to mind: "At previous companies I have used graphical test tools that were heavily invested in using coordinates to identify objects. This was a maintenance nightmare. Squish's Object Map is unique and greatly facilitates test readability, robustness, and maintenance."

In some cases, an object property's value may vary depending on the platform where the is application run. Squish's Object Map can accommodate such challenges since the properties used to identify an object can not only be matched for equality, but also using wildcard or regular expressions—a feature that Tim has found to be particularly useful.

In addition to Squish's documented features, Tim has found Squish's technical support team very helpful: "I was impressed with froglogic's support to quickly and efficiently help from the moment I started to evaluate Squish."

Conclusion

Perforce's Performance Lab depends on the reliability and repeatability of Squish tests to check application performance as well as behavior across multiple platforms. Squish's usefulness and flexibility has allowed Perforce to rapidly adopt Squish as an integral part of their performance quality monitoring process. This has lead to time and cost savings compared to the previous manual testing, while at the same time ensured tests are automatically and reliably repeated to ensure product quality.

Squish for Qt: Data Sheet

Squish for Qt is the market leading automated GUI testing tool for Qt applications, and is the only professional GUI testing tool available with dedicated support and a tight Qt C++ framework integration.

Squish for Qt uniquely offers the following features for automated GUI testing Qt applications on any platform or device:

- ☑ Dedicated C++ Qt GUI toolkit support

- ☑ Qt version 3.x, 4.x and 5.x support

- ☑ Desktop platforms: Windows, Linux, Unix and Mac OS X

- ☑ Embedded, Mobile and RTOS platforms: QNX, iOS, Android and embedded Linux

- ☑ Embedded native Windows control support (with Squish for Windows)

- ☑ Cross-platform test scripts without modification

- ☑ Script access to complete Qt APIs (beyond slots and Q_PROPERTYs)

- ☑ Script access to all slots and Q_PROPERTYs (also for custom controls)

- ☑ IDL mechanism for access to complete C++ API of custom classes

With dedicated support for:

- ☑ All standard Qt controls

- ☑ All complex Qt controls (item views, menus, tabs, etc.)

- ☑ QGraphicsView controls

- ☑ Embedded QtWebKit (when combined with Squish for Web)

- ☑ QML controls and QtQuick applications

- ☑ Custom controls derived from Qt controls supported

- ☑ Extension plugin mechanism (dedicated support for complex custom controls)

- ☑ and more...

Squish Customers

THOMSON REUTERS

SIEMENS

XILINX

IBM   hp

Google

Paradigm

AIRBUS

BOEING
Forever New Frontiers

Pfizer   ARM

SCANIA

DAIMLER

Shell