



mozilla
CORPORATION

Mozilla QA & The Tools We Use

Tony Chung, Marcia Knous, Carsten Book

July 29, 2008

Mozilla QA

7/29/2008



mozilla
CORPORATION

Overview

- I. Who We are
- II. Tools of the Trade
- III. How to Get Involved

Mozilla QA

7/29/2008



mozilla
CORPORATION

Who We Are

- I. QA Execution Team
- II. QA Test Development Team

Mozilla QA

7/29/2008



- QA Test Execution
 - Feature Test Planning and Ownership
 - Manual Test Case creation and execution
 - Bug and Data Analysis
 - Products: Firefox, Thunderbird, Distribution, Websites (AMO, mozilla.com, L10n)



- QA Test Development
 - Creation and maintenance of Test Tools
 - Create new Automated testcases for new features
 - Products: Gristmill, Mochikit, Reftests, XPCShell, Leak Testing, Accessibility, Mobile



mozilla
CORPORATION

Tools of the Trade

- I. Manual Testing
 - I. Litmus
 - II. Feedback Channels
- II. Automation Testing
 - I. Leaks Testing
 - II. Addons Testing

Mozilla QA

7/29/2008



- Introduction
 - Integrated testcase management and QA tool that is designed to improve workflow, visibility, and turnaround time in the Mozilla QA process
 - Used by both Mozilla QA and the community to report test case successes and failures



- Test Group Structure
 - Basic Functional Tests (BFT)
 - Full Functional Tests (FFT)
 - Smoketests
 - Specific Test Suites to operating systems such as “Windows Vista” or “Mac Keyboard Controls”
 - Localization I10n Test



- Test Subgroup Structure
 - Test Groups are divided into various subgroups.
Examples would be:
 - Tabbed Browsing, OS Integration, Installation, etc
 - QA team members help maintain the various Firefox subgroups and are continually adding and refining test cases as the product evolves from release to release.
 - We hope to get the community more involved in this process.



- **Hendrix** - <http://feedback.mozilla.org/>
 - Linked from the release notes and “What’s New” page
 - Many of the comments result in bugs being filed in Bugzilla
- **MozillaZine Forums** - <http://forums.mozillazine.org/>
 - Very useful discussions. Many thread discussions also turn into Bugzilla bugs.
- **Reporter** - <http://reporter.mozilla.org/>
 - Good for identifying evangelism issues as well as sites that were working but are now broken
- **Breakpad** - <http://crash-stats.mozilla.com/>
 - Crash reports information is aggregated here



mozilla
CORPORATION

Memory Leak Tests

- I. What is Leak Testing?
- II. The Tools Involved
 - I. Leak Gauge
 - II. Addons

Mozilla QA

7/29/2008



- **Lots of Memory Improvements for Firefox 3:**
 - The XPCOM cycle collector continuously cleans up unused memory. Plus, hundreds of memory leaks are now remedied
 - Memory Leak Tests after every Code change to make sure we introduce no new Memory Leaks



- QA also started with a lot of Tests for Memory Leaks
- Mozilla QA runs manual tests using the Litmus Suite
 - (using Tools such as Debug Builds and Leak Gauge)
- Regression Memory Leak Testing of Firefox
- Memory Leak Tests of new and popular Addon's
- Future: More Automated Tests run by Mozilla QA



- Various Tools for Performance and Leak Testing
- QA uses Debug Builds and Leak Gauge
- Debug Builds: Debug Builds can be build with Trace-Malloc Support to search for Memory Leak Builds every time up to date build with latest checkins
 - overview over Leaking Components (but Debug Builds need to be built individually and you need a Development Environment
 - Xcode, Visual Studio Express, etc) for most platforms
- Leak Gauge: Developed by David Baron –
 - It is designed to assist in detecting what leaks of large object graphs occur during normal browsing activity.
 - The logging can be run during normal browsing without significant overhead.
 - Log taken by setting environment variables in a release build.



- Using Firefox Debug Builds with Trace Malloc enabled for all Platforms
 - (Windows XP/ Vista, Linux, Mac 10.4/10/5)
- Trace Malloc allows searching Memory Leaks
 - Can provide a Log File Output
- Identify Leaking Components, this information helps Developers to debug and fix the Memory Leak!



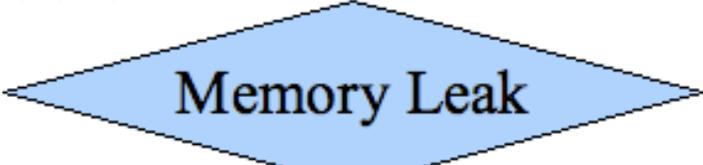
- Easy to use Memory Leak Test Tool developed by David Baron
- Only Requirement is a environment variable and works on all platforms
 - `c:\NSPR_LOG_MODULES=DOMLeak:5,DocumentLeak:5,nsDocShellLeak:5,NodeInfoManagerLeak:5`
- `set NSPR_LOG_FILE=c:\leak1.log` to define a Leak Log
- Works on Firefox 2 and Firefox 3+ Release Builds
- Leak Log Upload Form on <http://mxr.mozilla.org/mozilla/source/tools/footprint/leak-gauge.html> for analysis
- Upload Form indicates if a Memory Leak was found



- When a Memory Leak was found the output of the Log analysis look like this:

```
Leaked content nodes associated with node info manager at address 26a6ca0.
... with document URI
"jar:file:///C:/Program%20Files/Mozilla%20Firefox/chrome/toolkit.jar!/content/global/bindings/textbox.xml".
... with document URI "chrome://global/content/bindings/textbox.xml".
Leaked content nodes associated with node info manager at address 25a52e0.
... with document URI
"file:///C:/Program%20Files/Mozilla%20Firefox/extensions/yasearch@yandex.ru/chrome/content/yasearch.xbl".
... with document URI "chrome://yasearch/content/yasearch.xbl".
Leaked content nodes associated with node info manager at address 274c310.
... with document URI
"jar:file:///C:/Program%20Files/Mozilla%20Firefox/chrome/toolkit.jar!/content/global/bindings/checkbox.xml".
... with document URI "chrome://global/content/bindings/checkbox.xml".
Leaked content nodes associated with node info manager at address 27a8670.
... with document URI
"jar:file:///C:/Program%20Files/Mozilla%20Firefox/chrome/browser.jar!/content/browser/places/toolbar.xml".
... with document URI "chrome://browser/content/places/toolbar.xml".
Leaked content nodes associated with node info manager at address 28242e0.
... with document URI
"jar:file:///C:/Program%20Files/Mozilla%20Firefox/chrome/browser.jar!/content/browser/search/search.xml".
... with document URI "chrome://browser/content/search/search.xml".

Summary:
Leaked 0 out of 19 DOM Windows
Leaked 24 out of 112 documents
Leaked 1 out of 9 docshells
Leaked content nodes in 24 out of 124 documents
```



Memory Leak



- We test new Extensions uploaded to Addons.mozilla.org and also the Top-Downloaded and Recommend Addon's
- Leak Testing is done with Debug Builds and also with Leak Gauge
- We provide also a Leak Gauge “How-To” for Addon Developers and AMO-Editors
- A best practices Document is in progress to give Addon Developers a hand in avoiding Memory Leak
- We use new Profiles to avoid any false-positive Results from other Extensions



- We test all aspects of the Extension (Install, Uninstall, Features of the Extension)
- Provide Detailed Steps to reproduce so that this Memory Leak is reproducible by Developers and also to be able to verify the fix of this Leak.
- In case a Memory Leak was found we file a Bug in the Mozilla Bug Database and inform the Extension Developer
- **Future Plan: We will use more Automation to do this testing !**



- Successfully identified various types of Leaks in Extensions and Firefox Components – also in specific Scenarios – beyond automated Tests
- Memory Leak Logs from Debug Builds and Leak Gauge Help Developers to identify the cause of the Leak
- Memory Leak Testing of Extensions help to maintain the great Firefox 3 Performance



mozilla
CORPORATION

Getting Connected

- I. Bugdays & Testdays
- II. Betatesters alias

Mozilla QA

7/29/2008



- **Bug Days**
 - Weekly Triage of top unconfirmed bugs
 - Tuesdays, 7am – 5pm PDT, channel #bugday
- **Test days**
 - Focused testdays on Mozilla Feature testing
 - Topics examples: Firefox Nightlies, Feature areas, Test Case Writing
 - Fridays, 7am – 5pm PDT, channel #testday



- History of the betatesters alias:
 - Originally created to help web devs identify possible regressions in security and stability releases
- How it is used:
 - Email blasts are sent out when we take the product to the “beta” channel to give web devs a heads up to start their testing
 - Today the list contains over 4,000 members and is continually growing
 - Get more info here: http://wiki.mozilla.org/QA/Community/Betatesters_Mailing_List



- **Get Started!**
 - Visit <http://quality.mozilla.org/getstarted> for a simple tutorial
- Overview of Performance and Leak Tools
 - http://wiki.mozilla.org/Performance:Leak_Tools
- Memory Footprint Blog from pav
 - <http://blog.pavlov.net/2008/03/11/firefox-3-memory-usage/>
- More on QA Companion
 - <http://blog.mozilla.com/tomcat/2008/03/21/extension-memory-leak-testing>
- Contacts - qa@mozilla.org



- **Automation Firefox UI Testing (Thursday, 3:30pm, Alpine D)**
 - Automating Firefox UI actions through Gristmill as a means of accessible QA automation. [Clint, Mikeal, Adam]
- **Why I didn't write a test (Wednesday, 2:15pm, Alpine D)**
 - Stuff that's hard to test, and what we could do to fix it [ted mielczarek]
- **Web Performance Tools (Wednesday, 4:45pm, Alpine E)**
 - Performance and Memory Tools for the Web Platform [dbaron]
- **Bugzilla Lightning Talk (Thurs, 1pm, Emerald)**
 - Learn about Bugzilla Tips and Tricks [jesse]