

# Mozilla Release Engineering Brownbag for NewHires

(15feb2010)

# How to make your first code change

- **hg clone** <http://hg.mozilla.org/mozilla-central>
- Edit source files
- **hg diff -U 8 > foo.out**
- Attach foo.out to bug in bugzilla
  - set “r?”, “sr?” to ask for review
- If you have commit access, you can checkin a reviewed patch using “hg push”
  - “checkin-needed” to ask someone else to commit

# What happens when you checkin?

- Checkin to hg.m.o is detected
  - jobs to do builds are queued to run on next available machines
- Completed builds are put here
  - [http://ftp.mozilla.org/pub/firefox/tinderbox-builds/\[branch\]-\[OS\]/\[time\]](http://ftp.mozilla.org/pub/firefox/tinderbox-builds/[branch]-[OS]/[time])
- As builds become available:
  - jobs to run unittest are queued; results posted to tinderbox server
  - jobs to run performance suites are queued; results posted to tinderbox server and graphserver
- All with your change, and only your change.
- One checkin will trigger 110+ jobs
  - approx 40hours desktop and 25 hours mobile devices

# When you checkin: first we build...

- linux32
  - debug depend build; opt depend build
- linux64
  - debug depend build; opt depend build
- Win32
  - debug depend build, opt (PGO) depend build
- OSX 10.5
  - debug depend build; opt depend build
  - This universal binary is also OSX10.4 PPC compatible
- WinMO
  - debug depend build; opt depend build
- maemo4
  - debug depend build; opt depend build
  - three Fennec desktop build – one each for win32, linux, osx

# When you checkin: ...then we test!

- **linux32 opt:** mochitest (1, 2, 3, 4, 5, everythingelse), talos (tp4, nochrome, jss, dirty, cold , Tsvg, Tspider, Twinopen)
- **linux32 debug:** mochitest (1, 2, 3, 4, 5, everythingelse)
- **linux64 opt:** mochitest (1, 2, 3, 4, 5, everythingelse), talos (tp4, nochrome, jss, dirty, cold , Tsvg, Tspider, Twinopen)
- **linux64 debug:** mochitest (1, 2, 3, 4, 5, everythingelse)
- **Win32 opt:** mochitest (1, 2, 3, 4, 5, everythingelse), talos (tp4, nochrome, jss, dirty, cold , Tsvg, Tspider, Twinopen)
- **Win32 debug:** mochitest (1, 2, 3, 4, 5, everythingelse)
- **OSX 10.5 opt:** mochitest (1, 2, 3, 4, 5, everythingelse), talos (tp4, nochrome, jss, dirty, cold , Tsvg, Tspider, Twinopen)
- **OSX 10.5 debug:** mochitest (1, 2, 3, 4, 5, everythingelse)
- **WinMO opt/debug:** no tests yet
- **maemo4 opt:** mochitest (1, 2, 3, 4, 5, everythingelse), talos (tp4, nochrome, jss, dirty, cold , Tsvg, Tspider, Twinopen)
- **maemo4 debug:** mochitest (1, 2, 3, 4, 5, everythingelse)

# Then every night, we...

- do all the same builds again
  - full clobber, not depend
- create l10n repacks
  - 75 locales, per OS
- create “nightly” updates
  - someone running yesterday’s nightly build will be prompted to update to tonight’s nightly
  - partial update vs complete update
- run all the same unittest and talos suites



# Tracking it all: TBPL

- Organized by checkin/changeset
  - Easier to triage
- Only run when there is something to do
- Shared pool of identical machines

jruderman@mozilla.com – Sunday, December 06, 2009 5:03:20 PM -0800

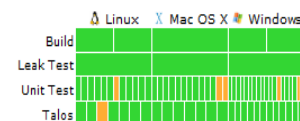
[3256d08b6705](#) Jesse Ruderman – Crashtest for [bug 509602](#)  
[a86c5f1d3fef](#) Jesse Ruderman – Add crashtests for several bugs

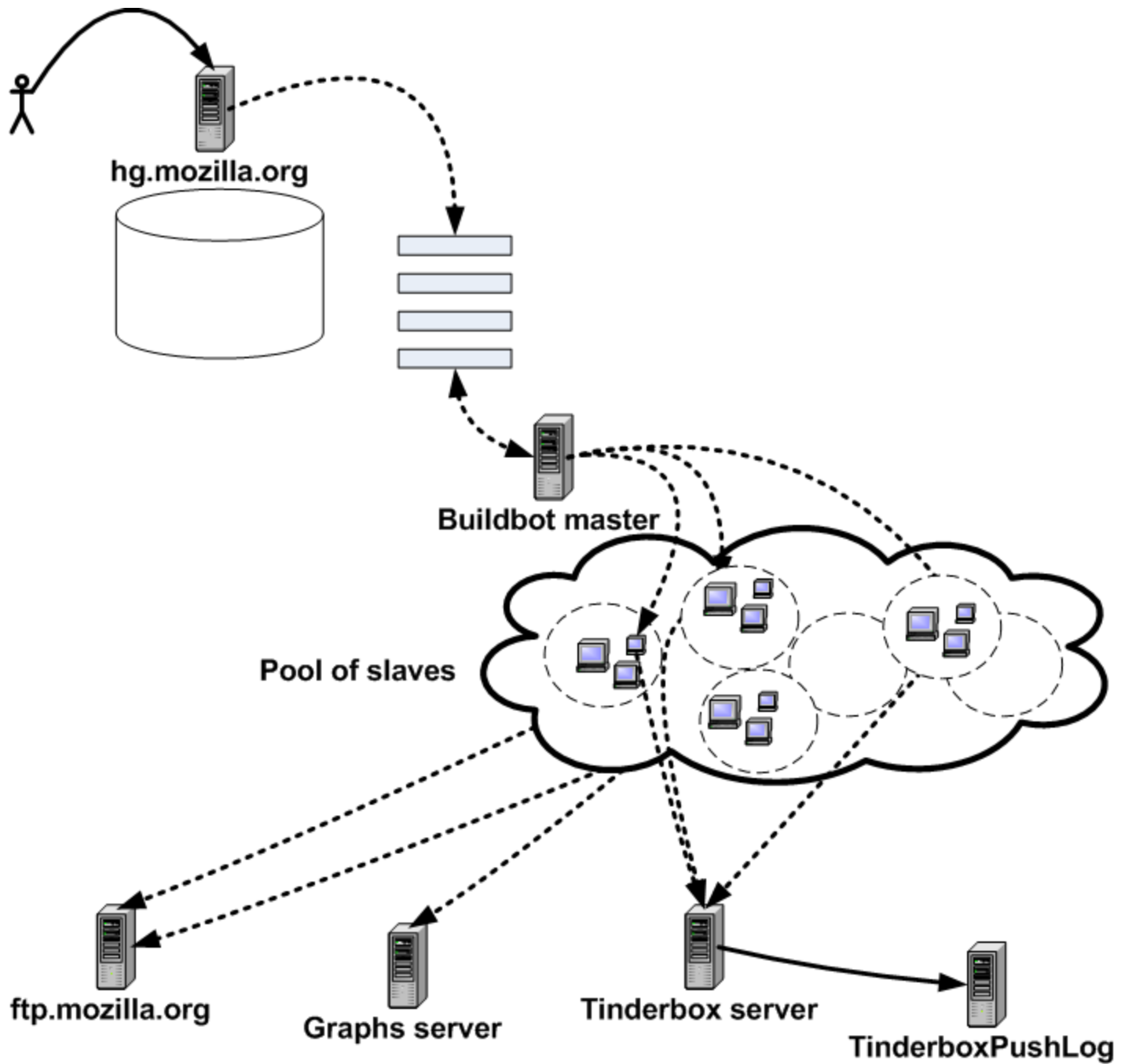
Linux	<b>B B L</b> Mo( <b>1 2 3 4 5</b> ) Md( <b>1 2 3 4 5</b> ) <b>Eo Ed T</b> T T T T T T
Mac OS X	<b>B L U M</b> ( <b>1 2 3 4 5</b> ) Mo( <b>1 2 3 4 5</b> ) <b>E Eo T T</b> T T T T T
Windows	<b>B B L U M</b> ( <b>1 2 3 4 5</b> ) Md( <b>1 2 3 4 5</b> ) <b>E</b>

vladimir@mozilla.com – Sunday, December 06, 2009 4:08:21 PM -0800

[7eb4406f8ded](#) Vladimir Vukicevic – [b=523959](#); [webgl] enable WebGL with EGL on Maemo and WinCE; r=stuart  
[f73d36907682](#) Vladimir Vukicevic – [b=532541](#); [webgl] drawElements needs to support ubyte as well as ushort i

Linux	<b>B B L</b> Mo( <b>1 2 3 4 5</b> ) Md( <b>1 2 3 4 5</b> ) <b>Eo Ed T</b> T T T T T T S
Mac OS X	<b>B L U M</b> ( <b>1 2 3 4 5</b> ) Mo( <b>1 2 3 4 5</b> ) <b>E Eo T T</b> T T T T T
Windows	<b>B B L U M</b> ( <b>1 2 3 4 5</b> ) Mo( <b>1 2 3 4 5</b> ) Md( <b>1 2</b> <b>3 4 5</b> ) <b>E Eo Ed T T T T T T T T T T T T</b>





# TryServer

- When you want to land your code, but you don't have build/test/perf machines running locally to first verify your code is safe to land on every OS
- ... or when you have experimental code you don't yet want to checkin, but you want to share tested builds on every OS with others
- For best results, refresh your local repo to latest *\*before\** using TryServer.
- Don't compare your results to other TryServer jobs.
  - Instead compare to results for production code you are based on – where your change is the only difference.

# To rerun specific tests on specific build

- Do file a bug in mozilla.org:RelEng and we will manually trigger
  - it used to not be possible to rerun on same build
  - RelEng working to make this self-service
- Do NOT just re-submit to TryServer
  - TryServer will rebuild everything, so your re-test will be on different bits.
  - It will take longer, wasting time running a bunch of others jobs you don't want.

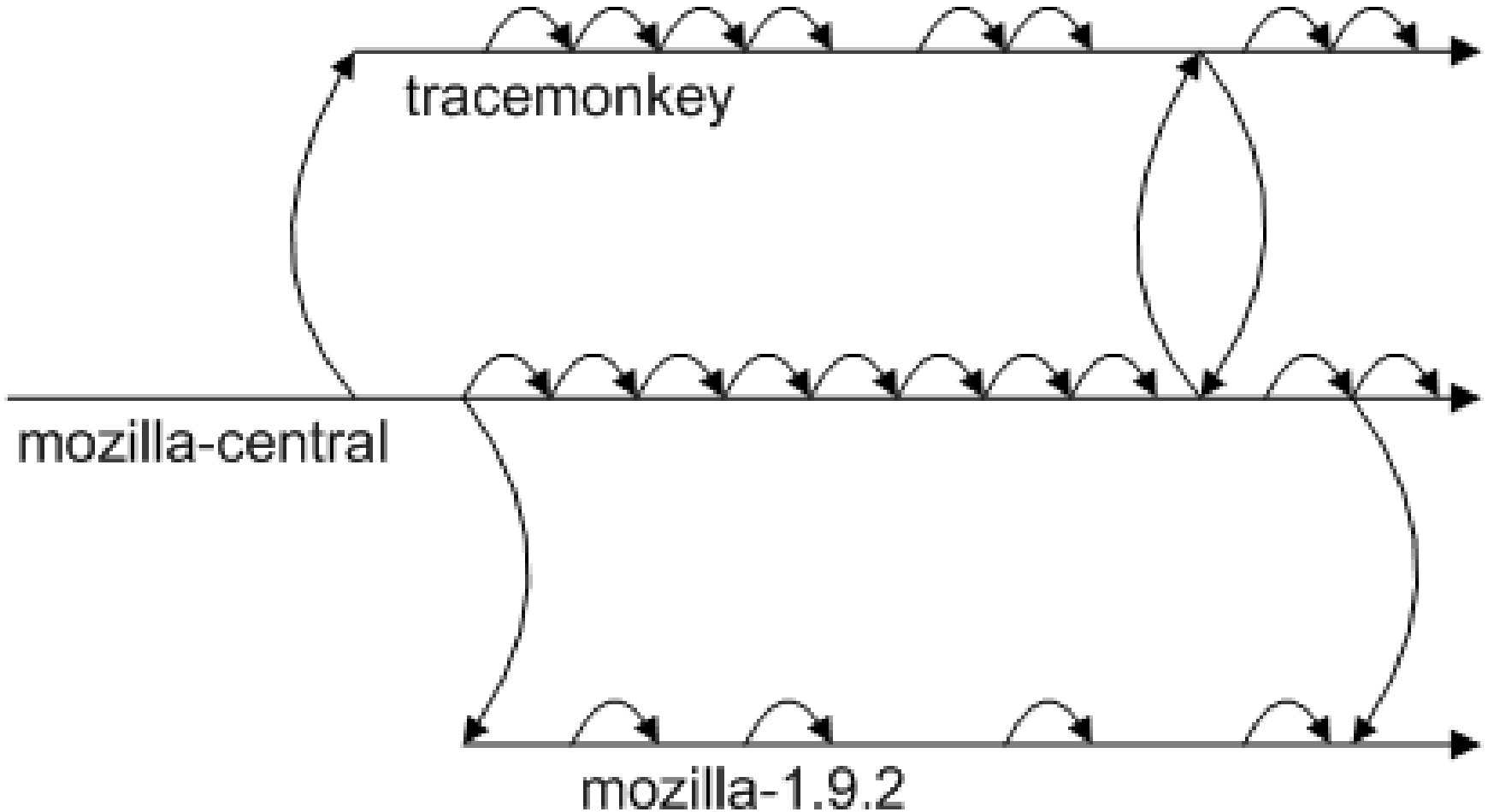
# Where and when to land changes?

- Using hg
  - mozilla-central, mozilla-1.9.2, mozilla-1.9.1, electrolysis, lorentz, places, tracemonkey, mobile-browser
  - if you need a new project branch, let us know
- Using cvs
  - mozilla-1.9.0, mozilla-1.8.1

# Where and when to land changes?

- If working on a project branch, land there first and make sure there are no regressions
  - i.e. tracemonkey
- Land on mozilla-central after checking with Sheriff and Tree rules
- If also needed in specific release, get approval in bug, and then land in mozilla-1.9.0/1/2 as directed by Release Driver.

# Where and when to land changes?



# Tracking work across branches

- File bugs for all work going to release branches, and get approval
- Use the correct branch flags!
- Followon work should be in linked bug.
  - Link to original bug
  - Use “regression” keyword
  - This helps release-drivers track regressions across branches, and helps prevent firedrill releases.

# Current terminology

- Tinderbox server / Tinderbox waterfall
- Tinderbox PushLog / TBPL
- Buildbot master, pool-o-slaves, pods
- TryServer
- Builds
  - depend, clobber, nightly, beta, RCs, release, debug, opt/PGO
  - L10N repacks, partner repacks
- Unittest suites
  - mochitest1-5, reftest, xpcshell, crashtest, browserchrome, a11y
- Talos/performance suites, GraphServer
  - Tp4, nochrome, jss, dirty, cold, Tsvg, Tspider, Twinopen
- Updates, AUS
  - partial, complete, nightly, beta, release, minor, major

# Current terminology

- Project branch
- “Tree stable timer”
- “Queue collapsing”
- “wait times”
- “end to end cycle time”
- Tree rules, “When is it ok to checkin?”
- “is the tree green”?
- Release Driver

# Obsolete terminology

- Tinderboxen / tinderboxes / tinderbox client
- Hourly builds
- Unittest machines
- L10n machines
- build-and-unittest

# Where can I get more help?

- Developers have “Sheriff of the day”
  - Schedule is on tinderbox
  - ReleaseEngineering now have “Builddduty” person to handle escalations
- Bugs in mozilla.org:ReleaseEngineering are best.
- #build in irc
- email [release@mozilla.com](mailto:release@mozilla.com)

Q&A