

389COM: Mozilla

Dr Carey Pridgeon, DR Nazaraf Shah

Coventry University

<http://www.coventry.ac.uk>

2016-08-06 Tue

Starting the process

- To begin with, get the Mozilla source code and finish it to do the initial setup (follow the accompanying Mozilla Build Worksheet).
- Once you've done this, and you have your Bugzilla account, use this lecture as a reference point.
- All subsequent Mozilla work on this module assumes you have completed the work sheet.
- Make completing it a priority, all questions will be answered *with have you done the Mozilla worksheet* first.

Choosing a bug - 1

- Mozilla have a nice website for choosing bugs to fix [bugs ahoy](#)
- When you select Mozilla product category, bugs in that category will appear.
- So now you refine this list.
- You can refine by language *C++*, *JavaScript* etc
- And choose to further refine by class of bug

Choosing a bug - 2

- simple (includes all good first bugs)
- Good first bugs are the place to start when learning the process.
- Unassigned bugs
- Diamond Bugs, like their namesake, are *very hard*.
- Choosing neither simple nor Diamond bugs would show all bugs.
- To gain any marks on a bug, simple or not, there must be at least evidence of an attempted fix.

Bug Fixing Process Guides

- [Mozilla New Developers Guide](#)
- [Mozilla Video Guides](#)

Creating a patch

- Patches create changesets that contain your source code changes.
- The command to create patches is `hg qnew patchName`
- The `hg` part is because this is a command in the `hg` toolset.
- This command will write a patch to the hidden folder `.hg/patches` in your source directory.

Editing the code - 1

- Before you start working on a patch, always update your source tree with `hg pull`
- Once you've edited your code to create a patch for submission, you need to clean your source code tree.
- to revert your code to a pre edit/pre patch creation state `hg strip tip -force`

Editing the code - 2

- If that doesn't work, just delete the source tree and set off a fresh clone. This isn't an ideal solution, but it is simpler than going through the other options.
- `hg clone https://hg.mozilla.org/mozilla-central/ firefox &`
- If you hit any other problem where patch creation, or pulling the latest revision won't work, try these as well.
- You will need to run mercurial setup again if you re-clone the codebase.

Bug Patch Submission on Nostromo - 1

- We can apply a patch to our clone on Nostromo test your bugfix, doing so will count towards your mark, but ultimately committing to Mozilla is the goal.
- Patches to be applied locally must be submitted in plenty of time.
- By this I mean not in the last week before submission.

Bug Patch Submission on Nostromo - 2

- In folder `/patches` on Nostromo, create a sub folder with your username.
- In that folder create sub folders for each patch, named with the patch id eg `1026865`, put in this the patch itself and a descriptive text file detailing what the patch is intended to do (typically the bug description from the bugzilla page).
- List the patch in [This Document](#)

Bug Patch Submission on Nostromo - 3

- Email ab0475@coventry.ac.uk with the patch number and your Username/Group Name in the email header, and the descriptive text in the body.
- Patches submitted at the last minute may not be applied and tested. *Do this as you go along, we will not rush at the end of the module to merge them.*
- Patches only count as being merged locally for assessment if you have an email from Nazaraf or myself saying it was merged and solved (or did not solve) the issue it was intended, to included in your patch submission portfolio.

Obligatory XKCD

SOMETIMES, WHEN I FIRST WAKE UP, I AM
CAUGHT IN THE HORRIBLE GRIP OF PERSPECTIVE:

IT MAY BE A JEWEL OF OPEN SOURCE, BUT
FIREFOX IS JUST A BROWSER. IT SHOWS
WEBPAGES. WHAT THE HELL IS WRONG WITH US?



FORTUNATELY, THE CURSORS CHANGE