****

**UNIVERSITI TUNKU ABDUL RAHMAN**

**FACULTY OF ENGINEERING & SCIENCE**

**ACADEMIC YEAR: 2015**

**MAY 2015 TRIMESTER**

**UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION**

**Group Project**

**MOZILA FIREFOX**

Course details

Course : Bachelor of Science (HONS) Software Engineering

Lecturer’s Name : Dr. Tay Yong Haur

Students’ Details

|  |  |  |
| --- | --- | --- |
| **Name** | **Student ID No.** | **Year/Sem** |
| 1. ANG ZI XUN | **1306976** | **Y2S3** |
| 1. ANG KEI YENG | **1302546** | **Y2S1** |
| 1. DANIEL KONG WENG KIT | **1405623** | **Y2S1** |
| 1. DING YING KEEN | **1102086** | **Y3S3** |
| 1. JAMES TAN TZE HWANG | **1400819** | **Y2S2** |

Contents

[Overview of the software we build 1](#_Toc429367038)

[Overall Configuration 2](#_Toc429367039)

[Modification did on Mozilla Firefox 3](#_Toc429367040)

[The process of Continuous Integration (CI) 7](#_Toc429367041)

[How new branches were handled 9](#_Toc429367042)

[Multiple developers working on the same project at the same time 16](#_Toc429367043)

[References 20](#_Toc429367044)

# Overview of the software we build

In this project, the software that we build is an open source web browser called “Mozilla Firefox” which is quite a well-known web browser among IT user. Mozilla Firefox is a free open source web browser developed by Mozilla Corporation and also many other volunteers. From the start of Mozilla Firefox, it is just a fork of Navigator Component of Mozilla Application Suite but after a period of time it had become the main development focus of foundation and official release which have replaced Mozilla Suite. However, nowadays Mozilla Firefox has become the second best browser in the world as reviewed on 20 February 2015. (Casserly, 2015) This browser has been the default browser of many windows user other than Google Chrome.

Mozilla Firefox is developed using several languages and web standard like HTML4, HTML5, XHTML, CSS, and C++ and so on. As on June 30, 2015, the newest version of Mozilla Firefox® 39.0 was released with many bug fixes, increase the speed of web browsing, improve standard compliance and implements new web application. (Mozilla Firefox, n.d.). The features of Mozilla Firefox web browser include tab browsing, bookmark, download manager, private browsing, and spell checking and so on.

However, the project we built today is actually the Mozilla Firefox that come from the GitHub website which has been modified in a repository called “gecko-dev” into Mozilla Firefox nightly and aurora.

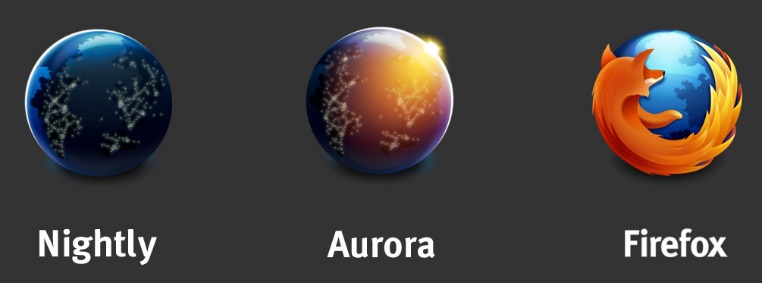


Figure 1 Different version of Mozilla Firefox

We try to download the codes from the repository and push to our own repository to modify on it to our own version.

# Overall Configuration

As stated in the overview above, this project had been found by us in a repository called “gecko-dev” of the project “Mozilla” in GitHub website ( <https://github.com/mozilla/gecko-dev> ).

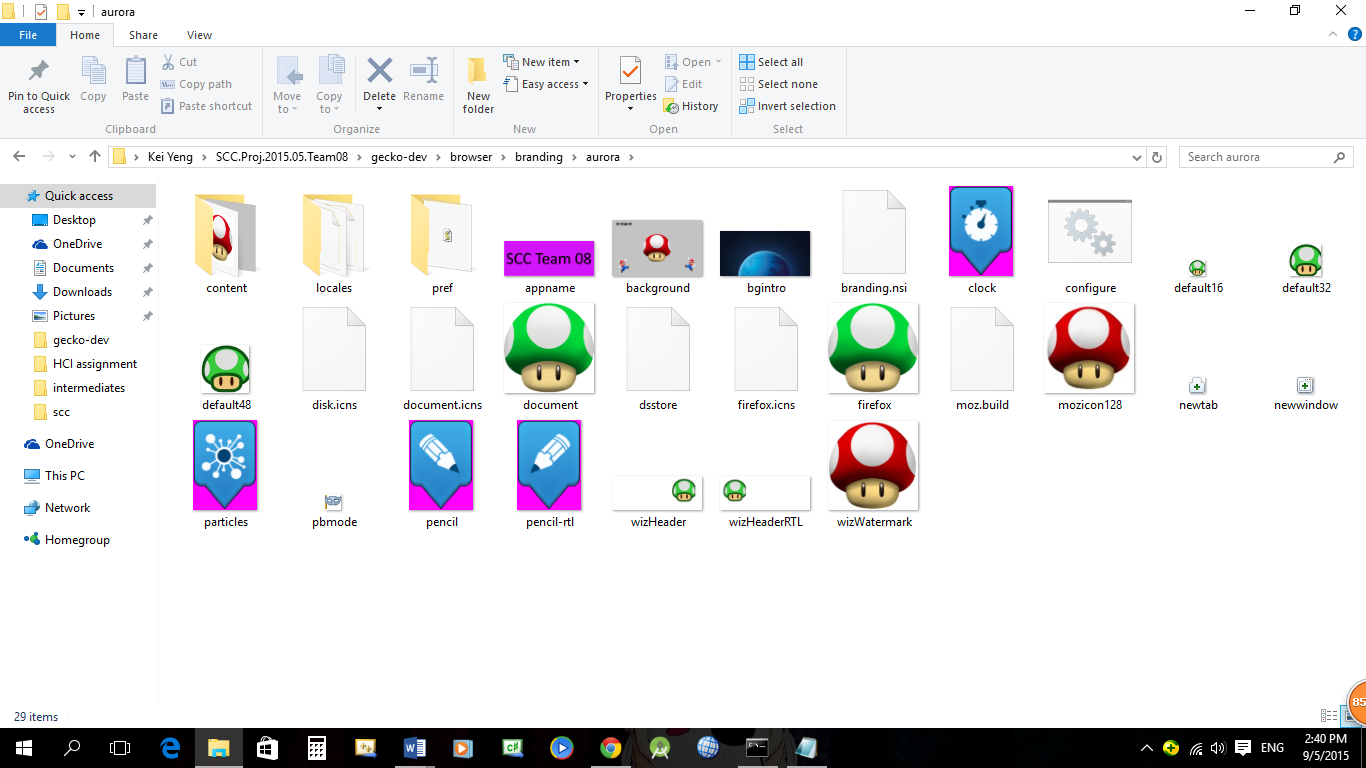
After found this project, we download the source code from there and then **git add –all, git commit –m “message”** and **git push** to our repository in GitHub which is a private repository called “**SCC.Proj.2015.05.Team08**” in “**UTAR-Projects**” created by Dr. Tay Yong Haur, our lecturer (<https://github.com/UTAR-Projects/SCC.Proj.2015.05.Team08> ).

Since we have pushed the source code to the GitHub, everyone started to **clone** from this repository to their GitHub desktop and start to modify on it. If someone modify it, we will use **git pull** to get the latest update from the repository and after we modified we used **git add --all**, **git commit** and **git push** to push our changes to the repository. However, keep doing it manually is too troublesome and so we use **Jenkins** as our server to do Continuous Integration (CI) so that it could help us to build the system each time we modify and push to GitHub periodically like check for updates every 2 hours and pull and every check-in it will be verified by an automated build so that we could found the problem in a short time.

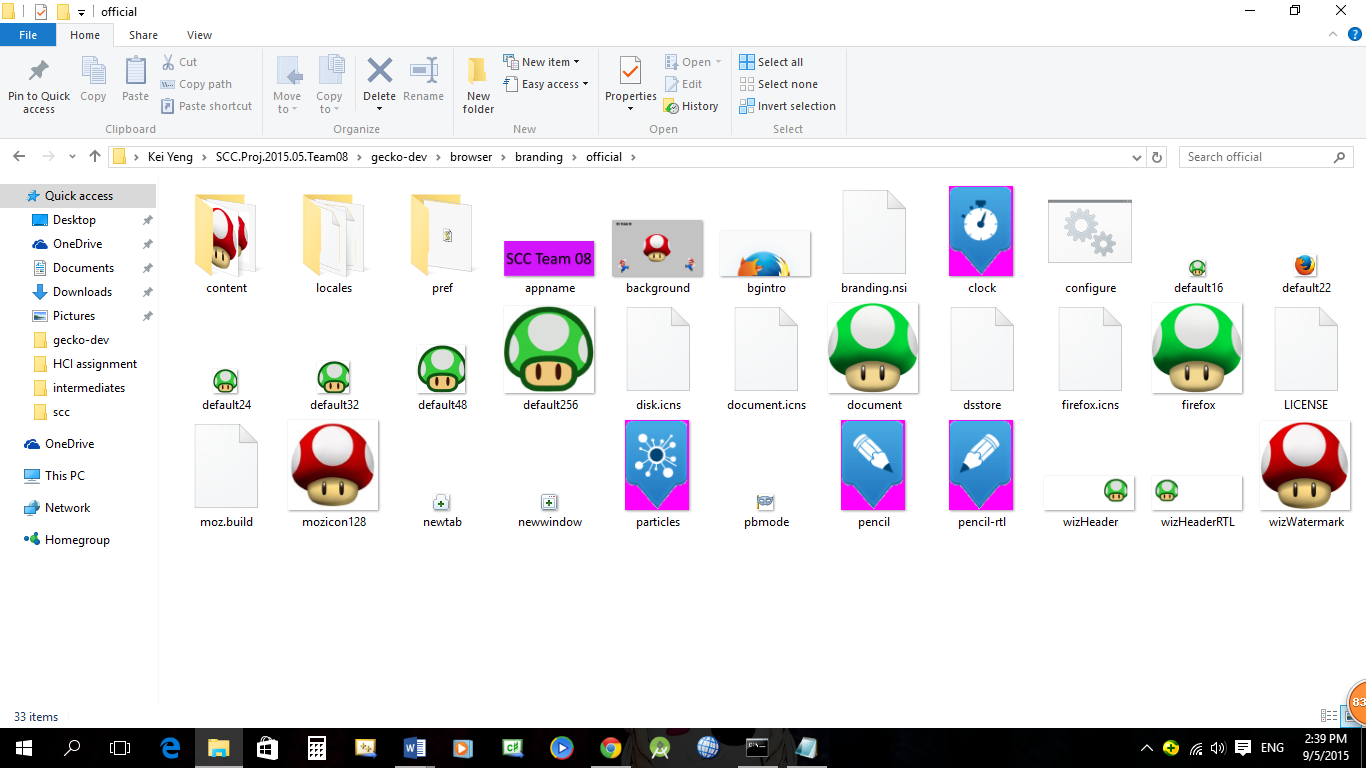
During doing the process of CI, our CI server is able to generate build history, static code analysis report and code coverage.

# Modification did on Mozilla Firefox

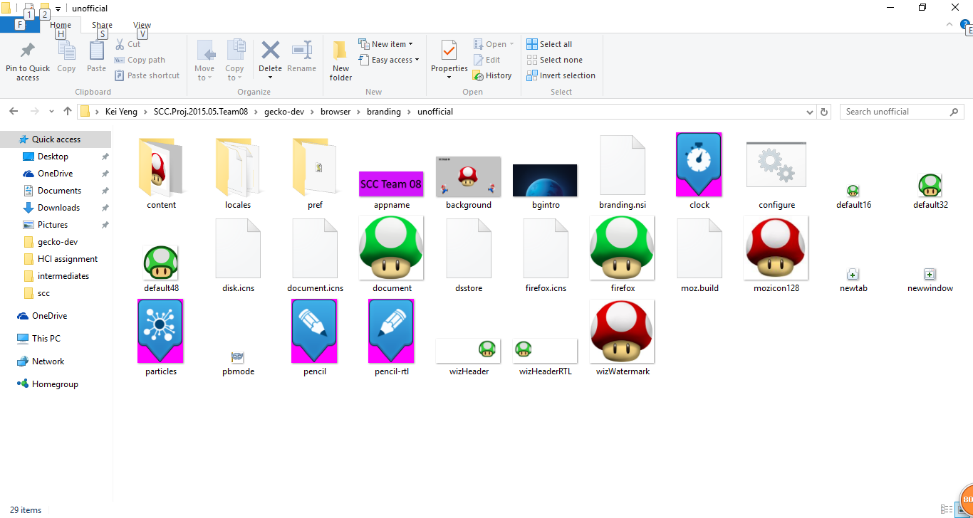
Changed all the icon to a popular video games Mario Bros’ mushroom character in C:\Users\USER\SCC.Proj.2015.05.Team08\gecko-dev\browser\branding\aurora



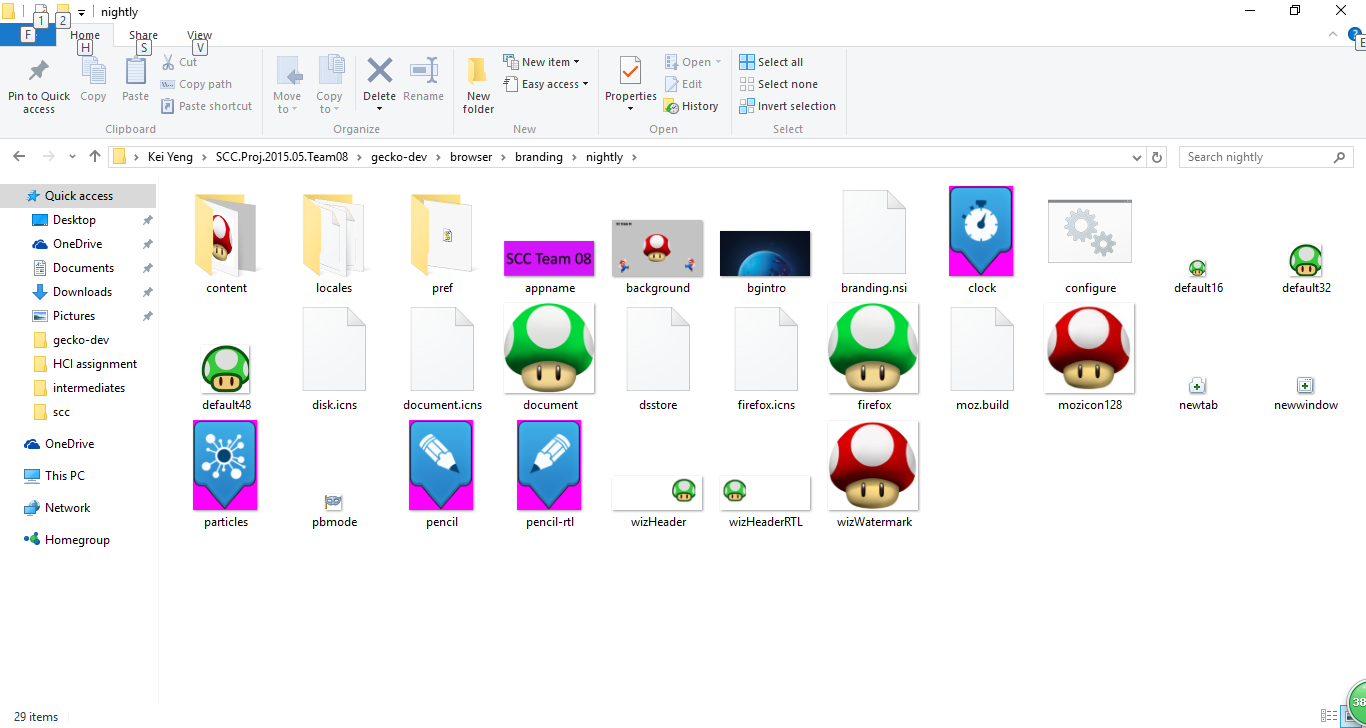
Changed all the icon to Mario Bros’ mushroom character in C:\Users\USER\SCC.Proj.2015.05.Team08\gecko-dev\browser\branding\official



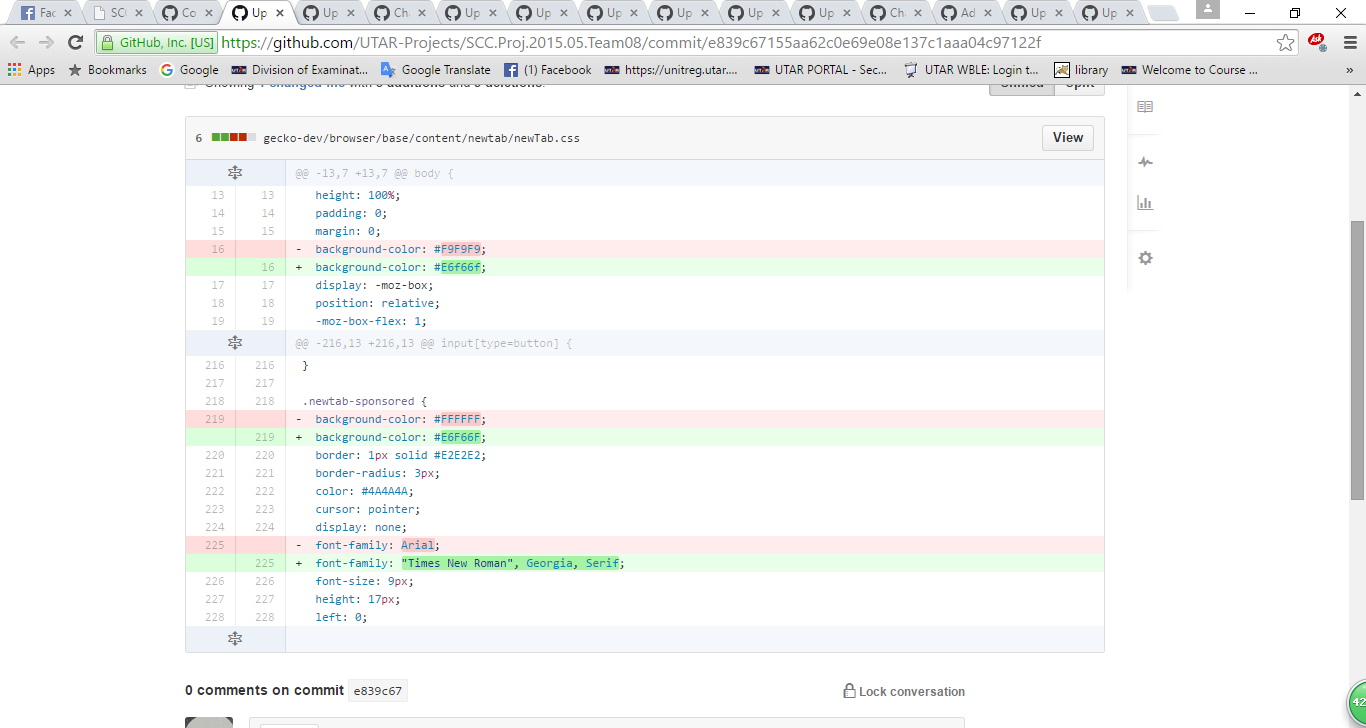
Changed all the icon to Mario Bros’ mushroom character in C:\Users\USER\SCC.Proj.2015.05.Team08\gecko-dev\browser\branding\unofficial



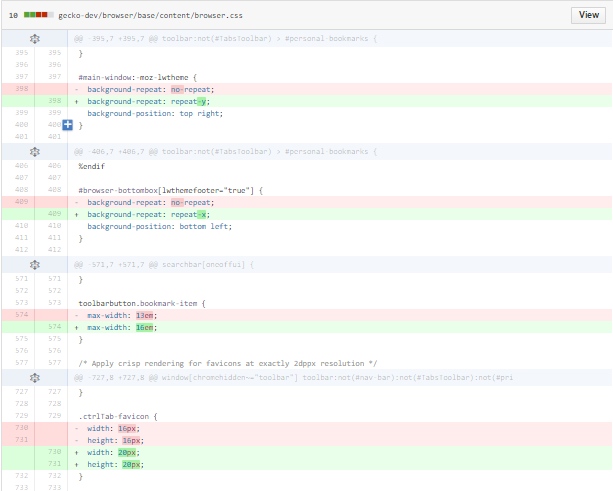
Changed all the icon to Mario Bros’ mushroom character in C:\Users\USER\SCC.Proj.2015.05.Team08\gecko-dev\browser\branding\nightly



Update newTab.css in gecko-dev/browser/base/content/newtab/newTab.css



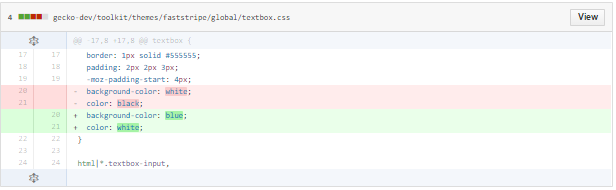
Update browser.css in gecko-dev/browser/base/content/browser.css



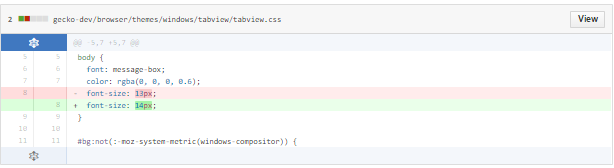
Update client.py gecko-dev/client.py



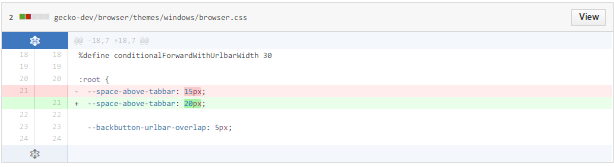
Changes in textbox.css for background-colour and colour in gecko-dev/browser/themes/windows/browser.css



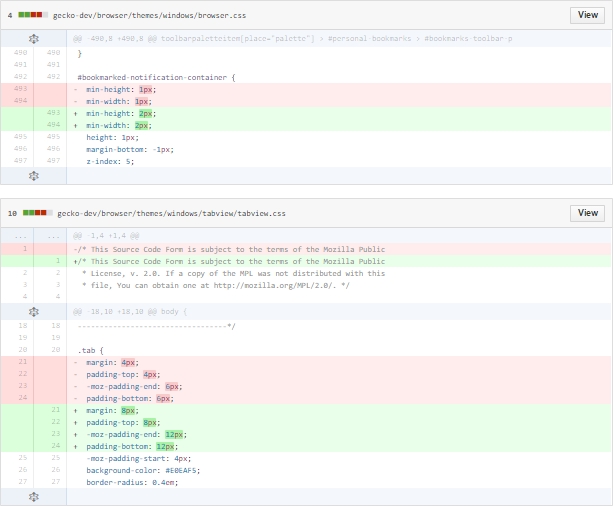
Update tabview.css in gecko-dev/browser/themes/windows/tabview/tabview.css



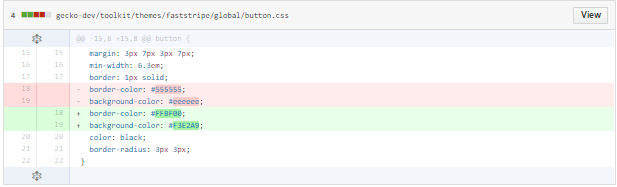
Update browser.css



Changes made in bowser.css in gecko-dev/browser/themes/windows/browser.css and tabview.css in gecko-dev/browser/themes/windows/tabview/tabview.css



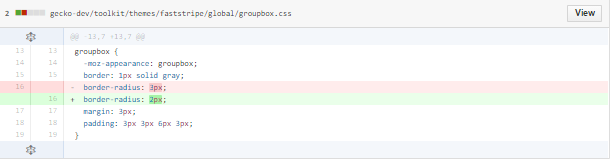
Update button.css in gecko-dev/toolkit/themes/faststripe/global/button.css



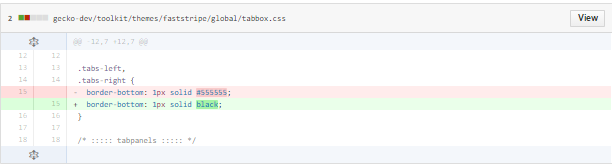
Update tabview.css gecko-dev/browser/themes/windows/tabview/tabview.css



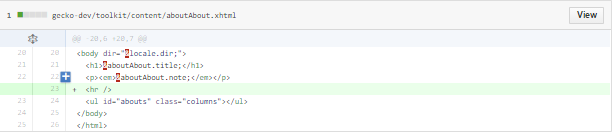
Update groupbox.css in gecko-dev/toolkit/themes/faststripe/global/groupbox.css



Update tabbox.css in gecko-dev/toolkit/themes/faststripe/global/tabbox.css

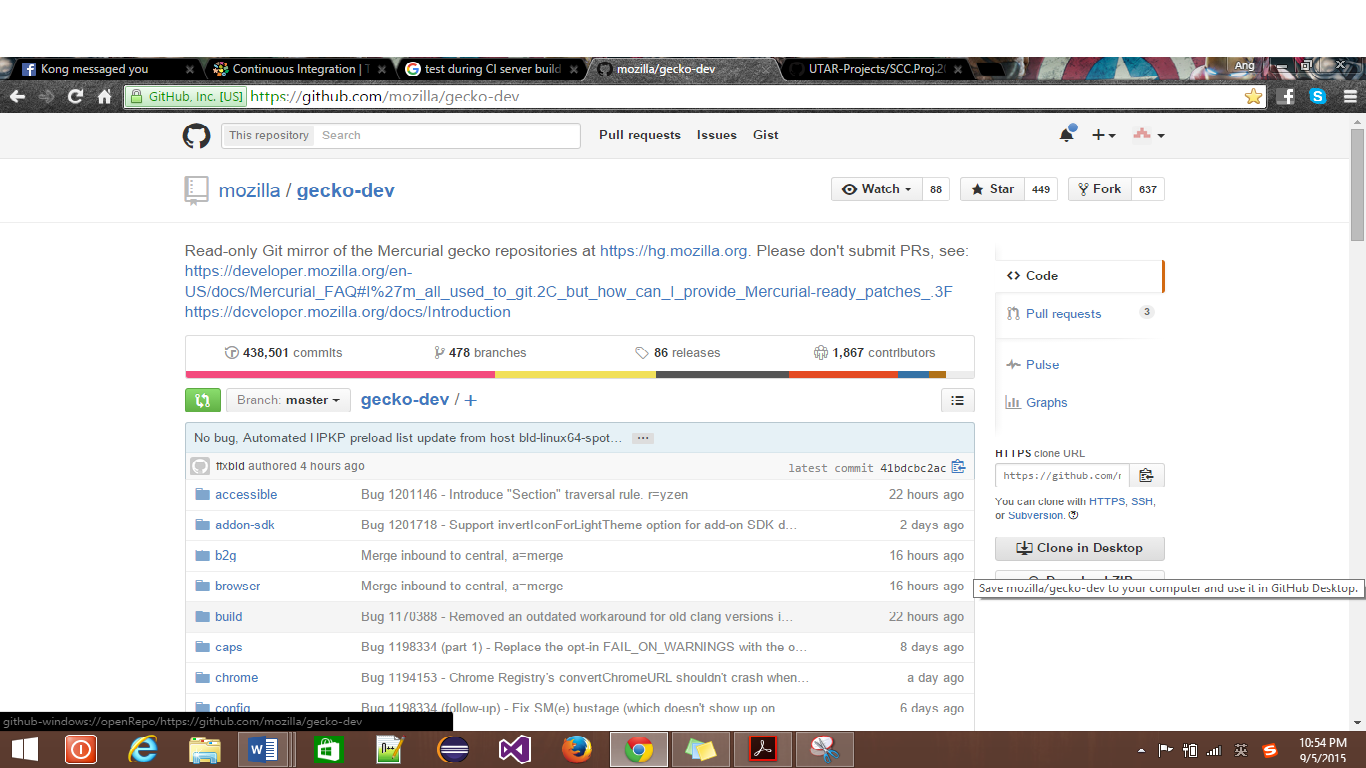


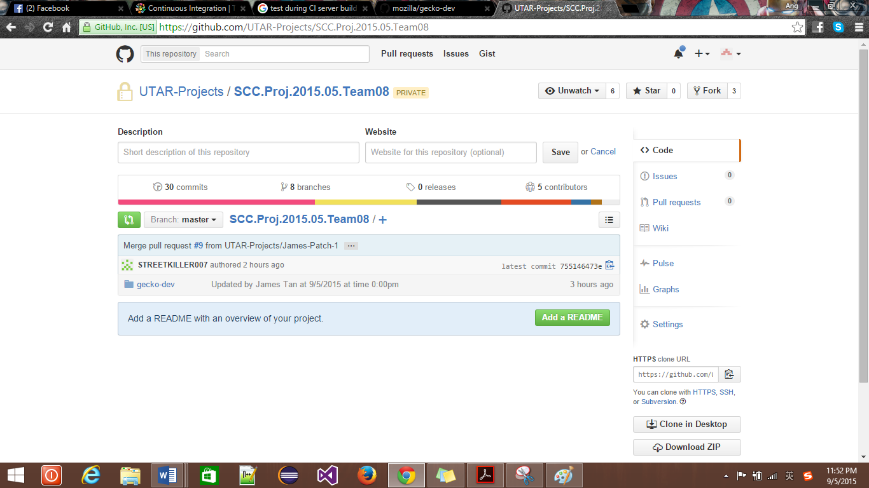
Added a horizontal line in aboutAbout.xhtml in gecko-dev/toolkit/content/aboutAbout.xhtml



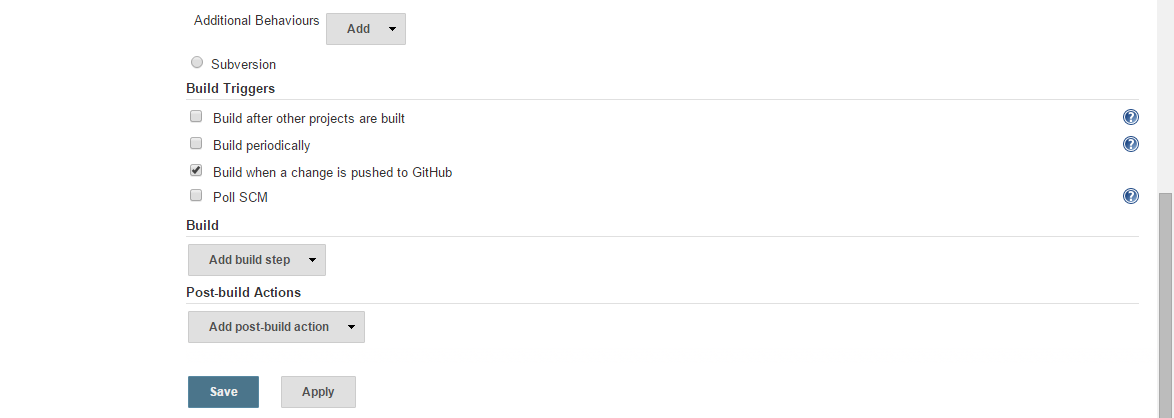
# The process of Continuous Integration (CI)

1. Clone the project code to our private repository “SCC.Proj.2015.05.Team08”

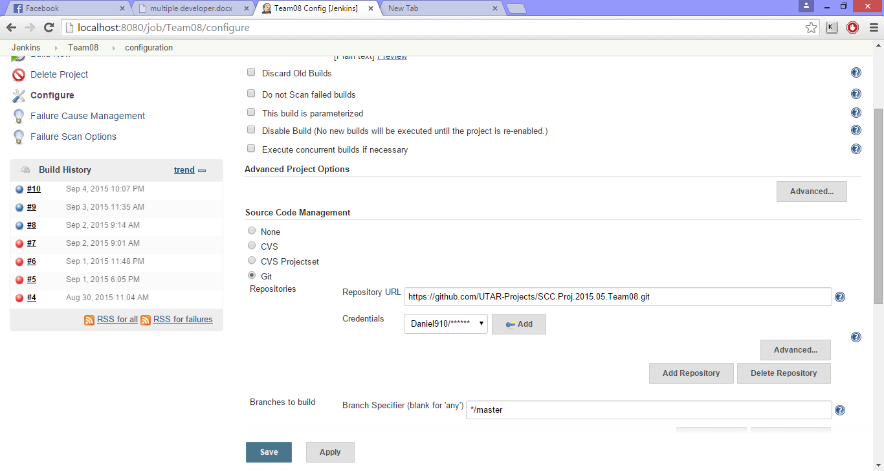




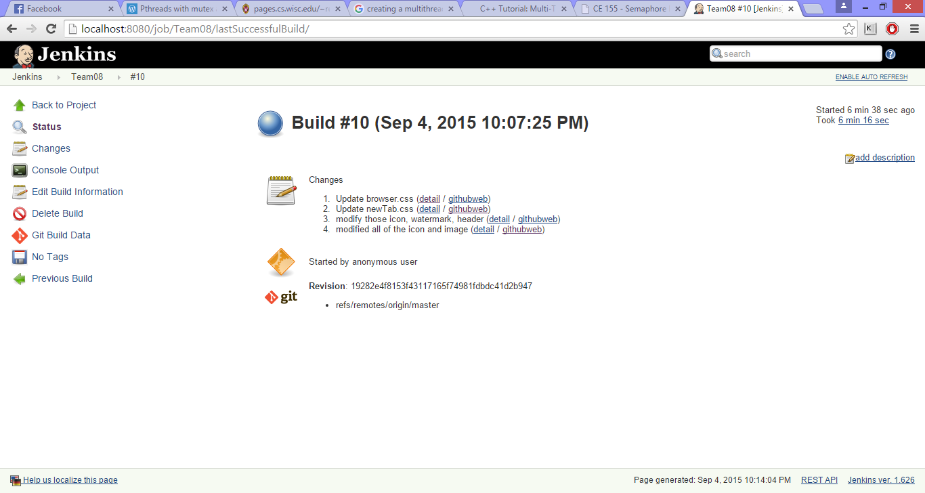
1. The CI server monitors the repository and checks out changes when they occur.

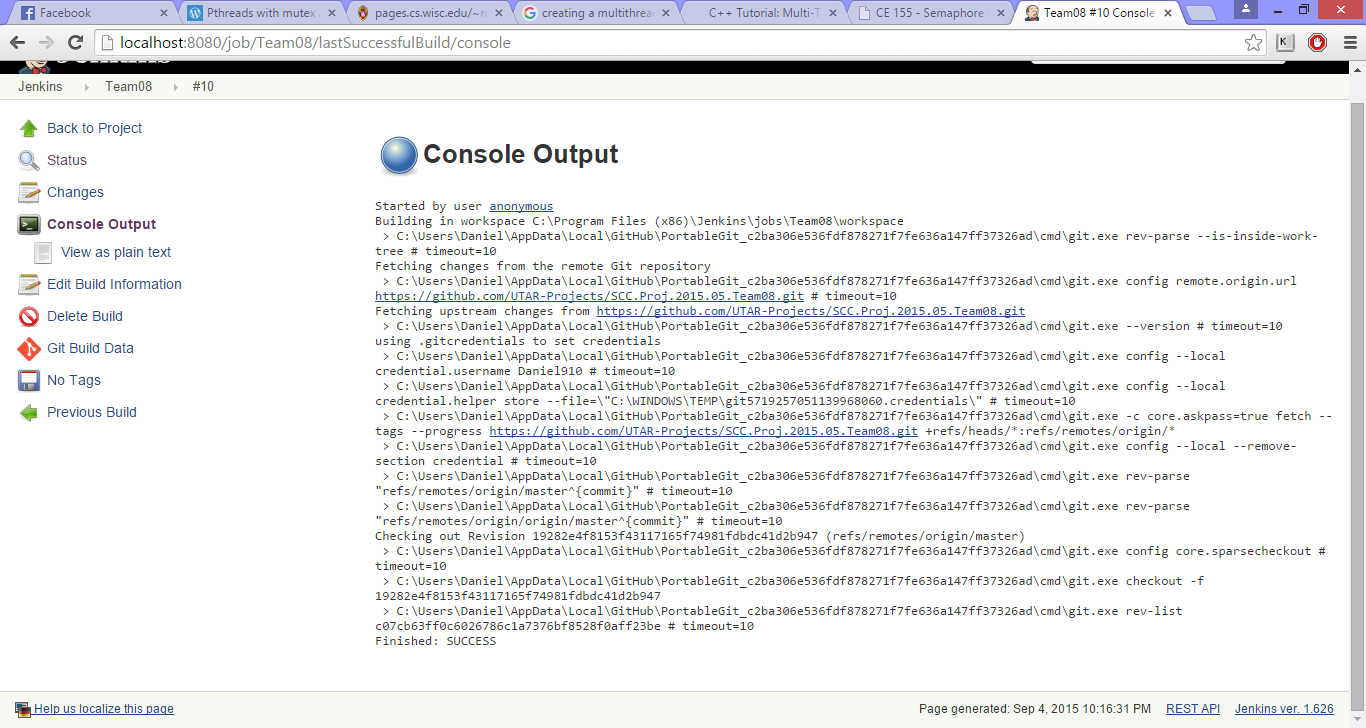


1. In Jenkins, before able to build the git project, one must create a job first. Under the job, we need to choose the source code management which will be git. And then, we have to specify the repository url as well as the credentials as the project is a private based. When all the required thing has been fulfilled, click onto the save and the new job is been created.

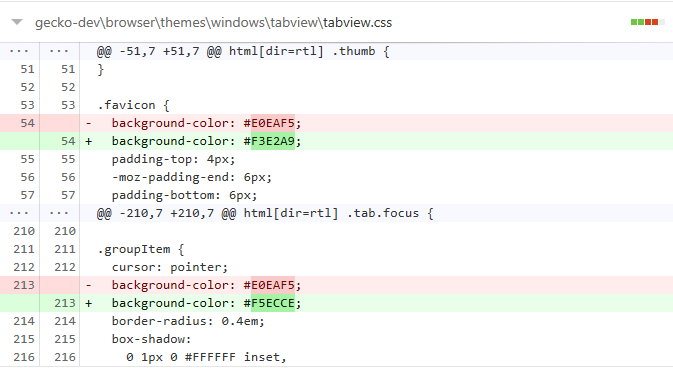


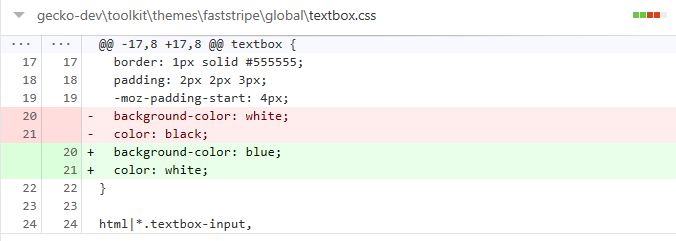
1. When job has been created, one can build the project now. The blue icon indicates the build is successful while the red icon states that it has failed. For our case, the project has been successfully built and it is on up to date with the latest commit made.



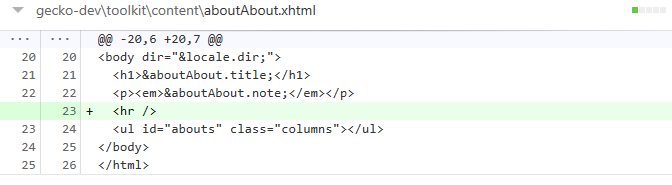
1. One can see the build in more detail by clicking on the console output. In here, it will provide a full description of your build. For instance, on which part does the project stopped to build due to error.

# How new branches were handled

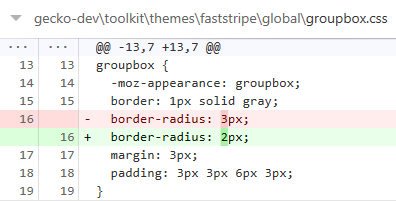
The branches that we created  
branch 1

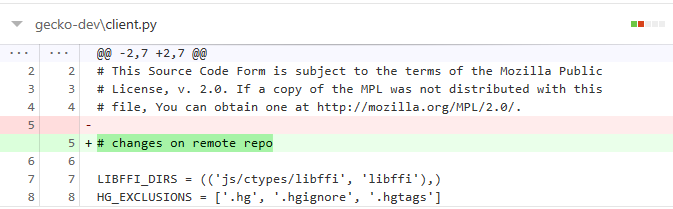
We had changed the background color of the tabview of the website.  
  
Branch 2

We had changed the textbox color to blue and the color to white.



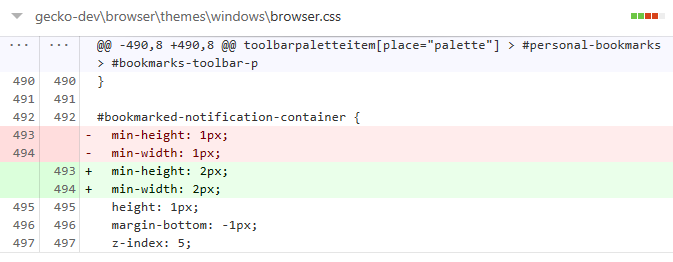
We had added a horizontal line.  
  
Branch 3



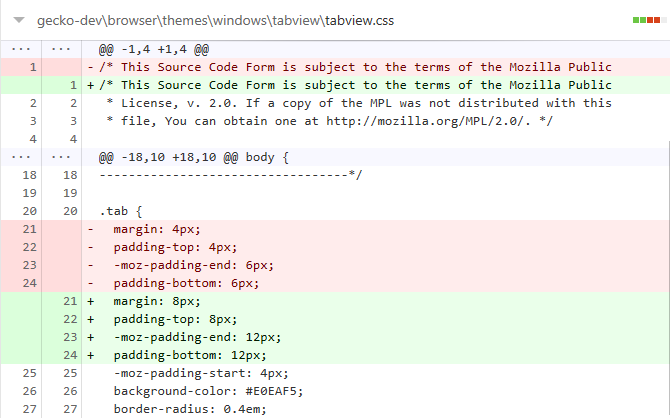
We had changed the border radius of the groupbox.

We had added in “changes on remote repo” in the client.py.

Branch 4

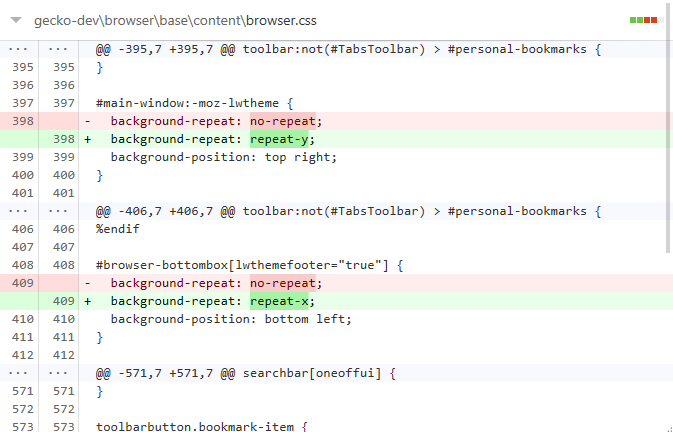


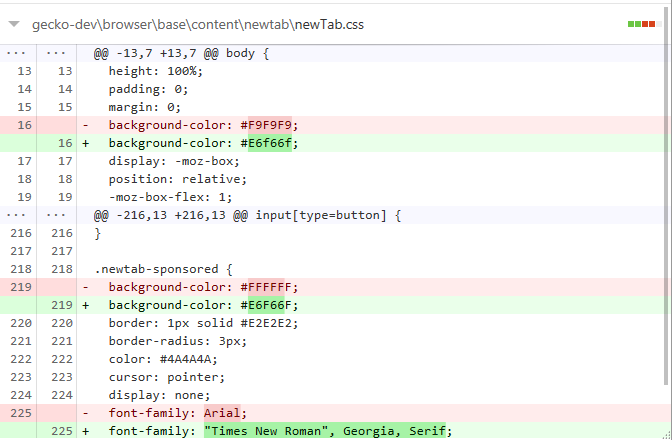
We had changed the minimum height and width to 2 pixel in the browser.



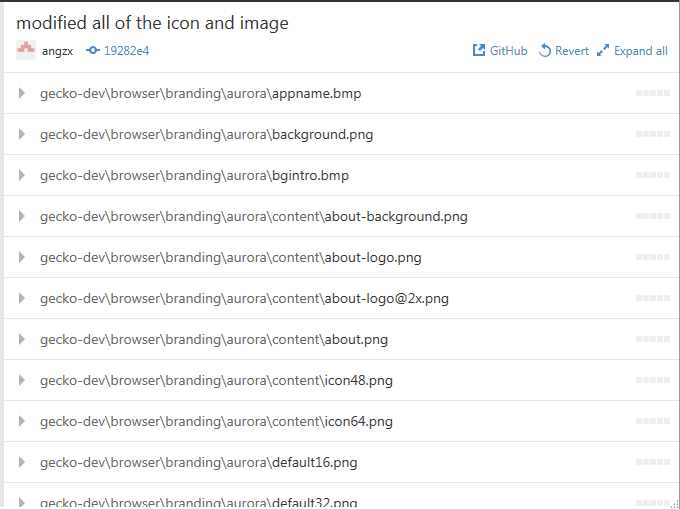
We had changed the value for margin, padding top, moz padding end, padding bottom.

Branch 5

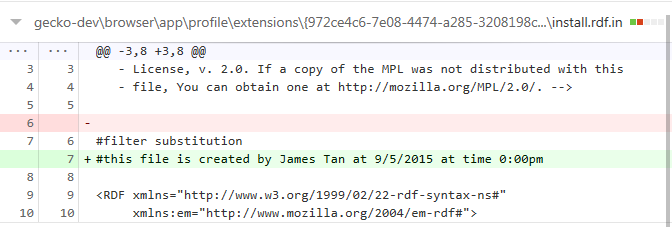


We had changed the background-repeat in the browser.

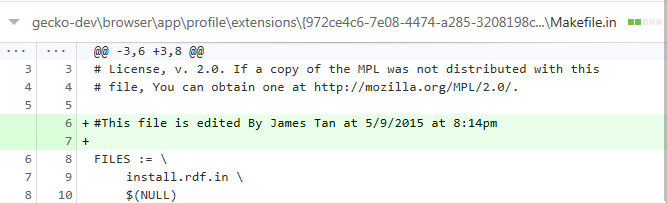
We had changed the background color, font-family in newTab.



We had modify the whole icon for software.

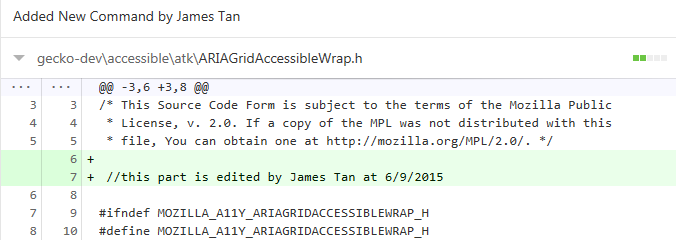
Branch 6

We had added in a new line of words and delete a line in install.rdf.in.



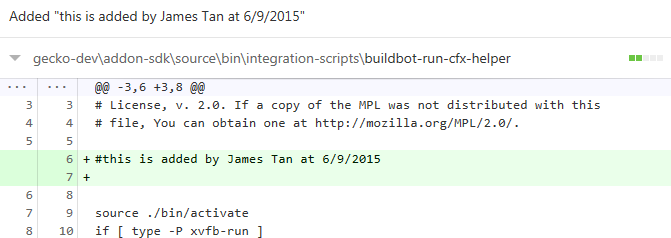
We had added in a new line of words and a blank new line in Makefile.in.

Branch 7



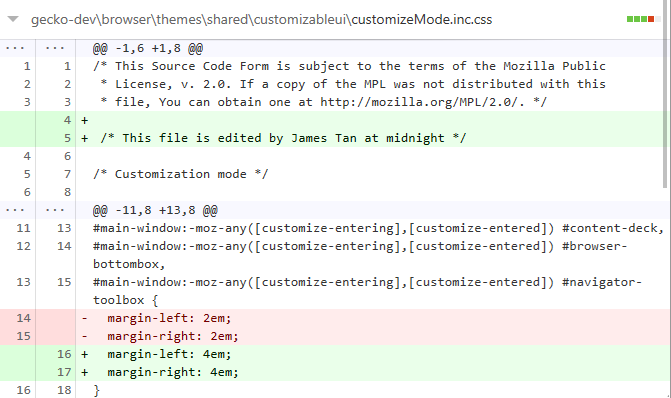
We add new line into the ARIAGridAccessibleWrap.h.

Branch 8

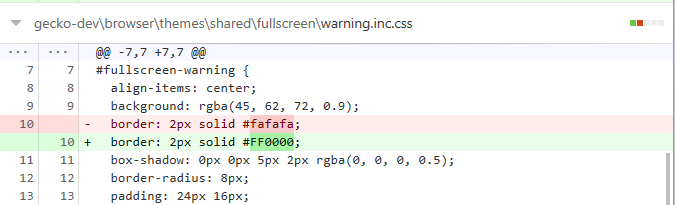


We had added a line of words and a blank line in buildbot-run-cfx-helper.

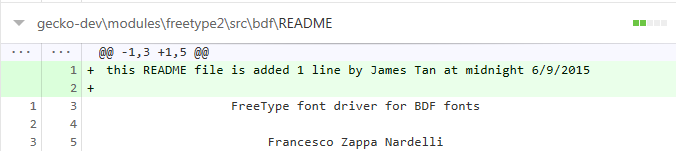
Branch 9



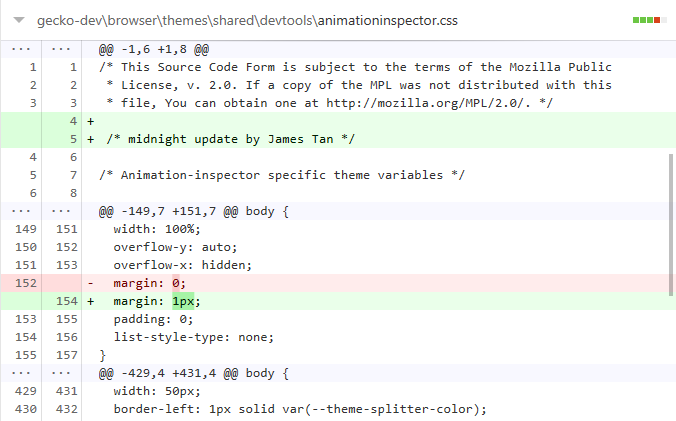
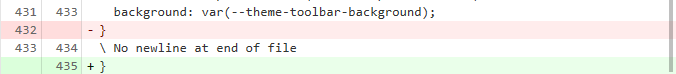
We had add a blank line, a line of words and changed the margin-left and margin-right from 2em to 4em in customizeMode.inc.css.



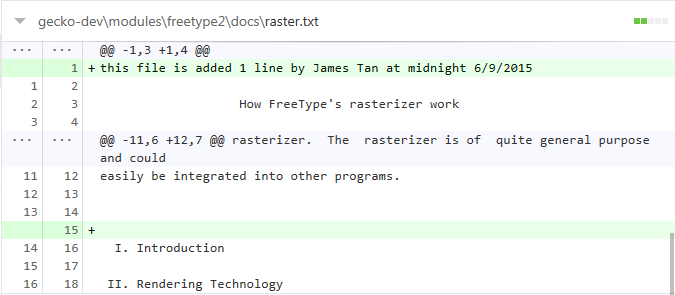
We had changed the color of the border in warning.inc.css.



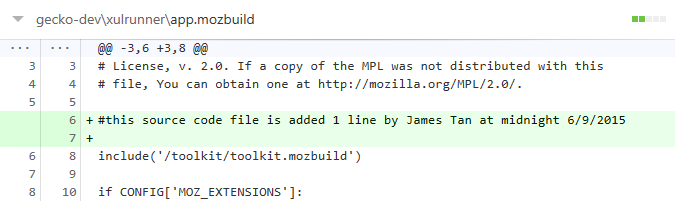
We had added a line of words and a blank line in README.

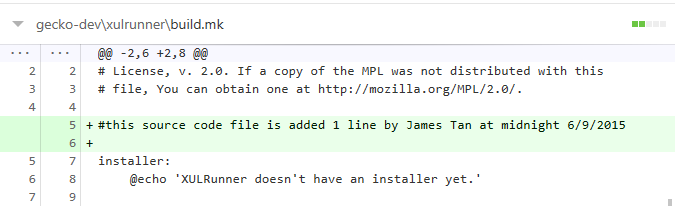
We had added a line of words, a blank line, changed the margin in body to 1px and move the bracket in animationspector.css.



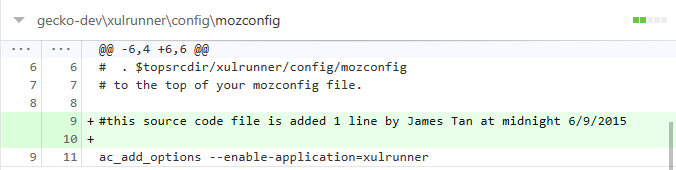
We had added a line of words and a blank line in raster.txt.



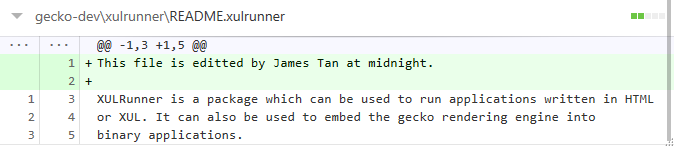
We had added a line of words and a blank line in app.mozbuild.



We had added a line of words and a blank line in build.mk.

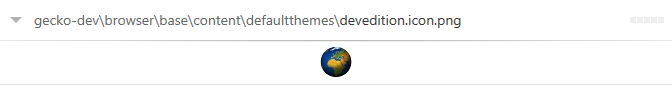


We had added a line of words and a blank line in mozconfig.



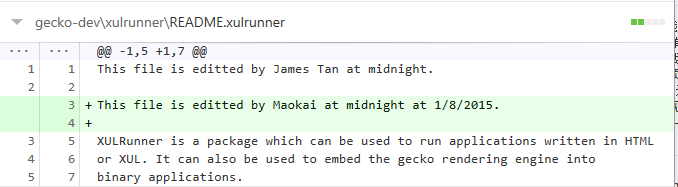
We had added a line of words and a blank line in README.xulrunner.

Branch 10



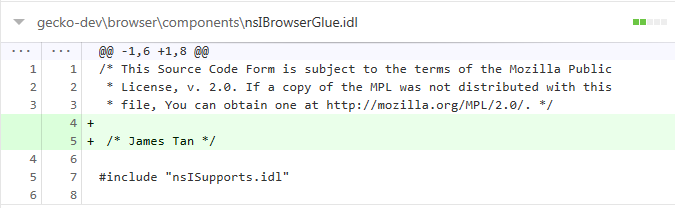
We had changed the devedition.icon.png to another picture in defaultthemes.

Branch 11

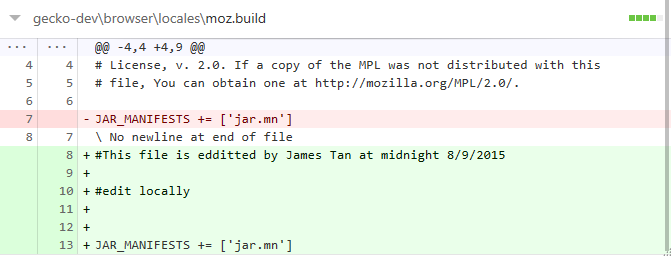


We had added a line of words and a blank line in README.xulrunner.

Branch 12



We had added a line of words and a blank line in nsIBrowserGlue.idl.



We had added 3 line of words, 3 blank line and removed 1 line of words in moz.build.

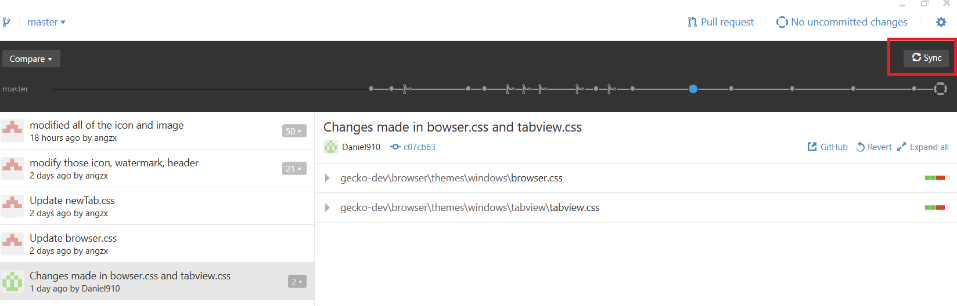
# Multiple developers working on the same project at the same time

After create a remote team repository with all the source code pushed on GitHub, other developers have to clone the repository for their own use by running “git clone” command follow by the repository URL so that all the work done by the team members can synchronize with each other. For GitHub, click the “clone in desktop” and it will automatically clone to GitHub desktop version.

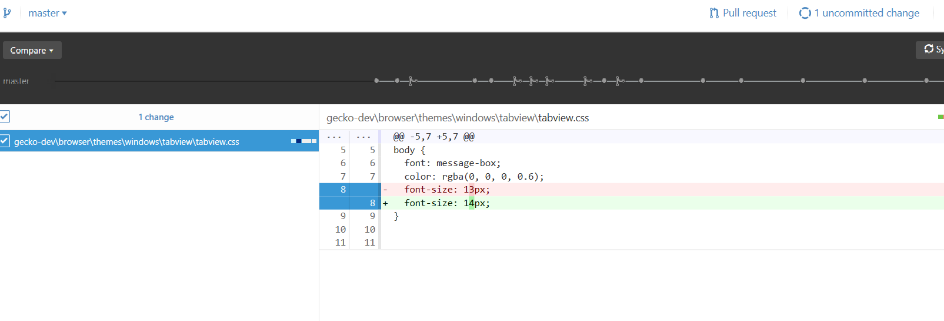
**Receive/Push Changes from/to the Remote Repository**

To get the latest changes that have been made on the remote server, the command “git pull” is used to retrieves the new commit/latest objects from the remote repository. After editing the code, run the command “git add –A” to add it all the modified file to the staging area (files are ready to be committed). After that, run the command “git commit –m ‘[message]’” with message to store the stage changes then “git push” to push the local repo to remote repository.

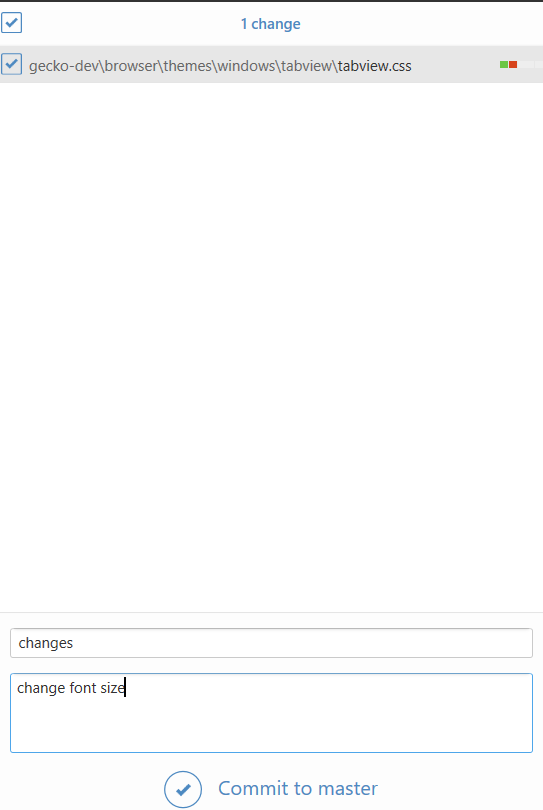
To get latest changes or to push local changes using GitHub desktop, press the “Sync” button on the top right corner that show in the figure below and GitHub will automatically pull or push changes to remote repository.



Any changes that make locally that haven’t commit, it will display as shown in the figure below.

****

To commit using GitHub, fill in the summary and description and press the “Commit to master/branch” button.



**Conflict**

If a project involve in multiple developers a good practice is performing git-pull, git-commit and git-push whenever finish a small part of function/form/design or other. This practice would help to get and update the latest version of the work and avoid conflict.

However, a conflict can still arises if the commit to be merged in has a change in another place and the current commit has a change in the same place. Git has no way of telling which change should take precedence. To resolve the conflict of commit, the only way is to edit the files to fix the conflicting changes. Then run “git add” to add the resolved files, and run “git commit” to commit the repaired merge. Git will remembers that you were in the middle of a merge, so it sets the parents of the commit correctly. For GitHub desktop, after fix the conflict, just commit and sync again like normal way.

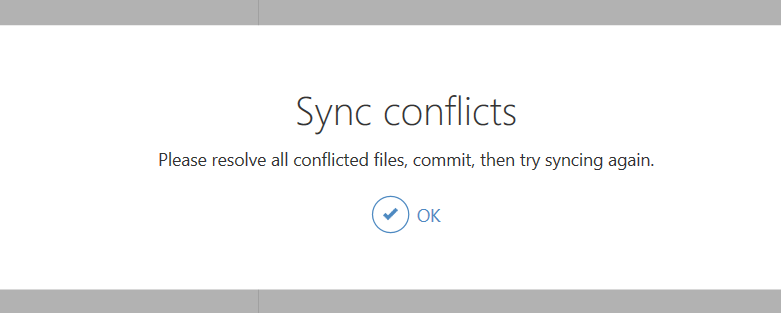


Figure above show the message from GitHub when pushing the conflict file to the remote repository.s

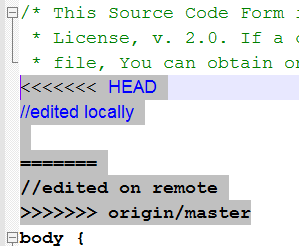


Figure above shows the line that conflict. To resolve this problem, first have to choose which version to keep and which version to delete. The version that edited locally is show between the   
<<<<<< HEAD

======

And the latest version on remote repository is show between

======

>>>>>> origin/master

Next, delete the “<<<<<< HEAD”, “======” and “>>>>>> origin/master” and also the version that don’t want to keep. After edit the file, continue to add and commit the file like normal way.

Between the process going, Jenkins is another program that we used to control this process cycle. Jenkins helps us to prevent integration problem and perform automated unit test while multiple developers are working at the same time. Another statement, building software project continuously, in the meantime, merging all developers working copies and checking integration problem continuously.

# References

Casserly, M., 2015. *Pc Advisor.* [Online]   
Available at: http://www.pcadvisor.co.uk/test-centre/software/which-is-best-web-browser-for-windows-2015-may-3493898/  
[Accessed 4 September 2015].

Mozilla Firefox, n.d. *Mozilla Firefox.* [Online]   
Available at: http://www.mozillafirefox.us/about-firefox.html  
[Accessed 5 September 2015].