CUNY Academic Commons - Feature #5434

Clean repository history

2016-04-11 10:30 PM - Boone Gorges

Status:	Resolved	**Start date:**	2016-04-11
Priority name:	Normal	**Due date:**	
Assignee:	Boone Gorges	**% Done:**	0%
Category name:	Meta	**Estimated time:**	0.00 hour
Target version:	Not tracked		

Description

The CAC git repository is humongous. The .git directory is almost 1GB in size. This makes it very difficult to work with: cloning takes a long time, and git status takes foooorrrreeeeevvvveeeerrrr the first time you run it during a session.

The root problem is that there are a number of large objects in the repository history. I think the most problematic ones are some fairly large MySQL zip files from 2009 or 2010, though there may be others. A side effect of the presence of these objects is that we are unable to make the CAC repo public, due to the non-public information in these zips.

I'd like to propose that I aggressively clean the history of the repository. This will mean that all existing clones will need to be recloned: all developer copies, cdev, and the production site. I can handle cdev and the production sites, but members of the development team would obviously be responsible for recloning their own repos.

The change will also mean that most existing hashes will be rewritten, breaking old links to Github changesets. This is obviously not ideal, but I think it's probably worth it. We can create an archived (private) copy of the existing repo in case we ever need to cross-reference old hashes.

I'm in the process of running some tests and some statistics to make sure that it's a worthwhile move, but I wanted to open this ticket first so that anyone with concerns could raise them here.

History

#1 - 2016-04-12 12:05 AM - Matt Gold

Thanks, Boone. To clarify, does this mean we'd be creating a public repo (something I'd be in favor of if possible)?

#2 - 2016-04-12 12:12 AM - Boone Gorges

It's a necessary prerequisite to creating a public repository. There is still other work to be done before it's possible, though (like offloading non-free plugins/themes to a build process).

#3 - 2016-04-12 12:19 PM - Raymond Hoh

Big +1 here!

Here's an article that might help:
http://stevelorek.com/how-to-shrink-a-git-repository.html

In that article, there is also a bash script to find out what the largest files in the repo are, but the script didn't work for me.

#4 - 2017-11-15 05:00 PM - Boone Gorges

- Status changed from New to Resolved

Some of this work has been done, the rest is probably diminishing returns.