

## SoundSoftware Code Site - Feature #1136

### Make repos available read-only through git as well as hg

2015-01-12 10:32 AM - Chris Cannam

<b>Status:</b>	New	<b>Start date:</b>	2015-01-12
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Chris Cannam	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			

#### Description

Because everyone wants git these days, and because it would be useful to be easily able to mirror projects to github, a nice and not too ambitious step would be to make the existing hg repo for each project be also available through git, in read-only form.

(If you want a read-write git repo, i.e. you want to host your project natively in git, you can do that already by e.g. hosting it at github and mirroring it here. Making this platform natively support both systems as alternatives is a possible future step, but at this point I'm most concerned about git access to projects that are already here in hg repos.)

In principle I think we could do it like this:

1. Clone each existing hg repo to a git repo in a sibling directory tree, and make the hg repo creation script also create this git repo
2. Serve the git version through https using the same authentication module as is currently used for hg, with the [git http backend](#)
3. Update the git repo automatically through a hook when something is pushed to an hg repo
4. List the git repo URL on the Repository tab for a project, marked as "read-only"

#### History

##### #1 - 2015-01-13 09:56 AM - Chris Cannam

A complication: committer names. Git requires a username and email address. Mercurial requires only "some text", and in many cases (specifically: almost all of mine) that text consists of a name but no email address.

So when I convert one of my repos using hg-fast-export I get a load of commits from Chris Cannam <devnull@localhost>.

In principle we have the information we need, though: each Redmine project has an optional set of mappings from committer name to project member (found in the project's Settings -> Repositories -> Users), and hg-fast-export [supports an authormap file](#). So we should be able to query the member username and email for each committer name on a given project and set up the authormap when converting.

The problem (besides accomplishing this, which is fiddly but not difficult) is that the mapping between committer name and project member may change over time. Indeed it's likely that a project member wouldn't even think to set up a mapping until they saw the results of converting to a Git repo -- e.g. I have several projects myself for which I haven't set up mappings. We certainly can't automatically regenerate the Git repo when the mapping changes, and adding a button to regenerate it manually would be a potentially hazardous complication.

That would be less of an issue if Redmine were less timid about creating default mappings. The help text in the Settings says "Users whose name or email matches that in the repository are mapped automatically", but in practice only users with matching emails appear to be mapped automatically.

##### #2 - 2015-02-11 04:45 PM - Chris Cannam

So as of commit:b31caaed9d4d, the default mapping in Redmine will map users who have matching names (but only if their firstname+lastname is unique across the site) in cases where email and login cannot be matched.

These mappings happen the first time a changeset object is created in the database, so we will need to go back and re-fill them for changesets

committed prior to, well, now.

Then the next step is to draw from that mapping to create a filemap when making a repo available via git.

### #3 - 2015-02-17 01:42 PM - Chris Cannam

I've now populated the user ids for those changesets whose user ids couldn't be resolved under the previous regime but can be resolved under the new one.

### #4 - 2015-05-19 05:38 PM - Chris Cannam

Added a script [here](#) which prints out an authormap file, but running this from the Rails runner is far too slow -- we should probably make it an api call.

### #5 - 2015-05-21 06:25 PM - Chris Cannam

And another script next to it (create-repo-authormaps.rb) which creates the authormap files. Once again though, it needs to be updated to use the api.

### #6 - 2016-01-12 05:57 PM - Chris Cannam

We now have an export-git.sh that does all of the actual repo export work. It looks good so far. Next up, serving the repos.

(Creating the authormap using the runner script turned out to be OK performance-wise, it takes 10-15 seconds for the c. 1000+ repos of which c. 290 public we have at the moment)

### #7 - 2016-01-18 01:05 PM - Chris Cannam

The script now creates bare repos in an appropriate place for them to be served as simple static files by Apache. So far, so good, and you can clone a repo via a URL like <https://code.soundsoftware.ac.uk/git/easyhg>.

One remaining problem: git on Ubuntu doesn't seem to know our CA root cert. It's fine on Arch Linux, OSX, and Windows 7 (the latter with SourceTree, I'm not sure whether different git clients get different cert stores).

### #8 - 2016-01-18 01:41 PM - Chris Cannam

The CA cert *is* shipped with Ubuntu:

```
$ openssl s_client -connect code.soundsoftware.ac.uk:443
[...]
Verify return code: 19 (self signed certificate in certificate chain)
$ openssl s_client -connect code.soundsoftware.ac.uk:443 -CAfile /etc/ssl/certs/AddTrust_External_Root.pem
[...]
Verify return code: 0 (ok)
$
```

It just isn't in /etc/ssl/certs/ca-certificates.crt.

**#9 - 2016-01-18 01:51 PM - Chris Cannam**

Hm no, that last command actually succeeds no matter which cert I specify with the CAfile option -- must have misunderstood the option. Or it misunderstood me.

**#10 - 2016-01-19 05:26 PM - Chris Cannam**

The answer, thanks to Dan: <http://stackoverflow.com/questions/7814423/ssl-works-with-browser-wget-and-curl-but-fails-with-git/16577227#16577227>

Reordering the certs in the intermediate chain on the server fixes it. The bundle provided to us gave them in the order root - intermediate 1 - intermediate 2; moving the root cert to the end was enough to make this work.

**#11 - 2016-02-05 03:55 PM - Chris Cannam**

OK, this seems to work quite well. The main problem now is that the repo mirror is only updated hourly. Could make this quicker, and/or have it driven by an hg push hook.