# Migration to GitHub

This page exists to flesh out the details of the migration to GitHub, which is being tracked under FCREPO-835.

**Contents:**



## Scope & Status

The following projects in the fedora-commons subversion repository on sourceforge.net will be moved:

| Subversion Directory | GitHub Repository | Status |
|---|---|---|
| fedora | github.com/fcrepo/fcrepo<br>github.com/fcrepo/fcrepo-before33* | Moved |
| services/diringest | github.com/fcrepo/diringest | Moved |
| services/genericsearch | github.com/fcrepo/gsearch | Moved |
| services/oaiprovider | github.com/fcrepo/oaiprovider | Moved |
| services/sipcreator | github.com/fcrepo/sipcreator | Moved |
| utilities/ | github.com/fcrepo/migration-2to3 | Moved |

** In order to make cloning and forking practical for the majority of contributors, the fcrepo repository will contain everything from the 3.3 release forward. The fcrepo-before33 repository will contain everything prior to 3.3. Instructions will be provided to locally graft the repositories for those who are interested in having the full view of history in a single repository.*

The projects in the incubator directory will be kept as-is.

## Migration Steps

The following steps were followed for the dry run migration of the fcrepo repository. Similar steps will be followed for the final migration of each repository (steps 4-6 apply to the fcrepo repository only).

### 1. Run the Initial Conversion

Because git stores a name and email address per commit, the conversion needs to know this information for each subversion userid.

First, we need the list of all subversion userids:

```
svnadmin dump /path/to/svnrepo | grep -a -A 2 svn:author \
| grep -v svn:author | grep -v ^V | grep -v ^- | sort -u > authors.txt
```

Next, this file will need to be modified so that each line looks like:

```
userid = Full Name <email-address@example.org>
```

Now the initial converstion can be run with git-svn. It takes a very long time to run (about half a day), so should be run with screen or nohup, like this:

```
nohup git svn clone file:///path/to/svnrepo --prefix=svn/ \
--authors-file=authors.txt --trunk=fedora/trunk --tags=fedora/tags \
--branches=fedora/branches 1.initial-conversion&
```

## 2. Abandon Subversion

Use a copy of the result of the previous step:

```
cp -a 1.initial-conversion 2.abandoned-svn
cd 2.abandoned-svn
```

Install the git-svn-abandon scripts, then run the following:

```
git svn-abandon-fix-refs
git svn-abandon-cleanup
git config --remove-section svn
git config --remove-section svn-remote.svn
rm -rf .git/svn .git/logs/refs/remotes/svn .git/refs/remotes/svn
```

## 3. Clean up Refs

Use a copy of the result of the previous step:

```
cp -a 2.abandoned-svn 3.refs-cleaned
cd 3.refs-cleaned
```

Remove useless tags:

```
git tag -d NA1-0 2.2.4-patch1 release-2.2.4-patch1
```

Install git-retag, then rename and sign original release tags using git convention (preserves original tag metadata):

```
git retag fedora-1-0 v1.0
git retag fedora-1-1 v1.1
git retag fedora-1-1-1 v1.1.1
git retag fedora-1-2 v1.2
git retag fedora-1-2-1 v1.2.1
git retag fedora-2-0 v2.0
git retag fedora-2-1 v2.1
git retag fedora-2-1-1 v2.1.1
git retag fedora-2-1b v2.1b
git retag fedora-2-2 v2.2
git retag release-2.2.1 v2.2.1
git retag release-2.2.2 v2.2.2
git retag release-2.2.3 v2.2.3
git retag release-2.2.4 v2.2.4
git retag release-3.0 v3.0
git retag release-3.0b1 v3.0b1
git retag release-3.0b2 v3.0b2
git retag release-3.1 v3.1
git retag release-3.2 v3.2
git retag release-3.2.1 v3.2.1
git retag release-3.3 v3.3
git retag release-3.3.1 v3.3.1
git retag release-3.4 v3.4
git retag release-3.4-RC1 v3.4-RC1
git retag release-3.4.1 v3.4.1
```

Remove old, redundant tags:

```
git tag -d release-1.0
git tag -d release-1.1
git tag -d release-1.1.1
git tag -d release-1.2
git tag -d release-1.2.1
git tag -d release-2.0
git tag -d release-2.1
git tag -d release-2.1.1
git tag -d release-2.1b
git tag -d release-2.2
```

Remove old, unused branches (all branches but those specifically excluded below will be removed):

```
git branch -D `git branch | grep -v master \
| grep -v maintenance-3.4 | grep -v maintenance-2.2 \
| grep -v fcrepo-604 | grep -v fcrepo-644 \
| grep -v fcrepo-748 | grep -v fcrepo-756 \
| grep -v fcrepo-775 | grep -v fcrepo-579 \
| grep -v fcrepo-586`
```

Free unused space in the repository:

```
git gc --prune=now
```

Avoid 'local uncommitted change' errors later

```
git status
```

## 4. Separate Recent History (fcrepo)

Use a copy of the result of the previous step:

```
cp -a 3.refs-cleaned 4.recent-history
cd 4.recent-history
```

Remove tags and branches prior to 3.3

```
git tag -d `git tag|grep -v 3.[34]`
git branch -D maintenance-2.2 fcrepo-579 fcrepo-586
```

Find the id of the 3.3 tagged commit and parent:

```
export lastid=`git show v3.3|grep ^commit|sed 's/...... \(.*\)$/\1/'`
export parentid=`git show $lastid^|grep ^commit|sed 's/...... \(.*\)$/\1/'`
```

Filter out everything prior:

```
git filter-branch --parent-filter "sed -e 's/-p $parentid//'" \
--tag-name-filter cat -- --all ^$parentid

git for-each-ref --format='%(refname)' refs/original \
| while read ref; do git update-ref -d "$ref"; done
```

Re-sign the tags:

```
git retag v3.3 v3.3
git retag v3.3.1 v3.3.1
git retag v3.4 v3.4
git retag v3.4-RC1 v3.4-RC1
git retag v3.4.1 v3.4.1
```

Reclaim unused space:

```
git reflog expire --expire=0 --all
git repack -ad
git prune
```

Find the new id of the 3.3 tagged commit (now the root commit of this repository):

```
export lastid=`git show v3.3|grep ^commit|sed 's/...... \(.*\)$/\1/'`
```

## 5. Separate Ancient History (fcrepo-before33)

Use a copy of the result of step 3:

```
cp -a 3.refs-cleaned 5.ancient-history
cd 5.ancient-history
```

Remove recent tags, branches, and commits:

```
git tag -d `git tag|grep 3.[34]`

git branch -D `git branch | grep -v master \
|grep -v maintenance-2.2 | grep -v fcrepo-579 \
|grep -v fcrepo-586`

git reset --hard $parentid
git gc --prune=now
```

## 6. Graft Test

Use a copy of the result of step 4:

```
cp -a 4.recent-history 6.graft-test
cd 6.graft-test
```

Copy the pack and index files from fcrepo-before33:

```
cp ../5.ancient-history/.git/objects/pack/* \
.git/objects/pack
```

Create the graft:

```
echo $lastid $parentid > .git/info/grafts
```

Copy historic tags and branches:

```
cat ../5.ancient-history/.git/packed-refs >> .git/packed-refs
```

Test it (all history, including old tags and branches, should be visible):

```
gitk --all
```

## 7. Push to GitHub

After creating a project at GitHub:

```
git remote add origin git@github.com:fcrepo/projname.git
git push origin master
git push --all
git push --tags
```