Table of Contents

Release procedures for frontier-related software.................................................................1
  Client--------------------------------------------------------------------------------------1
  Servlet-------------------------------------------------------------------------------------1
  Tarballs------------------------------------------------------------------------------------1
    frontier-squid tarball-------------------------------------------------------------------1
    frontier-awstats tarball-----------------------------------------------------------------2
  RPMs---------------------------------------------------------------------------------------2
Release procedures for frontier-related software

The Frontier system has many releasable pieces to it, and here are the procedures for each piece for developers to use.

This is what is on this page:

**Client**

To be moved here, but for now at http://frontier.cern.ch/dist/FrontierProjectNotes.html

**Servlet**

To be moved here, but for now at http://frontier.cern.ch/dist/FrontierProjectNotes.html

**Tarballs**

**frontier-squid tarball**

1. Update the SRC and RELEASE variables in squid/Makefile
2. Set an environment variable for the new version, for example:

   FSQUIDVER=4.4-1

3. If it is a new upstream squid version, do these steps

   ```
   SQUIDVER="`echo $FSQUIDVER|sed 's/-.*//g'`"
   cd ~/src/frontier/tarballs/frontier-squid4/distrib
   git rm squid-*.tar.gz
   rm -rf squid-*
   wget http://www.squid-cache.org/Versions/v4/squid-$SQUIDVER.tar.gz
   wget http://www.squid-cache.org/Versions/v4/squid-$SQUIDVER.tar.gz.asc
   git add squid-$SQUIDVER.tar.gz*
   gpg --verify squid-$SQUIDVER.tar.gz.asc squid-$SQUIDVER.tar.gz
   cd /tmp
   tar xzf $OLDPWD/squid-$SQUIDVER.tar.gz
   cd squid-$SQUIDVER
   scl enable devtoolset-2 "./configure --disable-wccp"
   scl enable devtoolset-2 make
   ```

   Then compare `/tmp/squid-$SQUIDVER/src/squid.conf.documented` to
   squid/files/postinstall/squid.conf.documented and update it and the corresponding
   squid.conf.proto with the things that have changed (at minimum, the version number will always
   change). Even if the upstream squid version hasn't changed, just the release number, update the
   release number in squid.conf.proto.

4. Add notes in the RELEASE_NOTES file for the new version.
5. Commit all changes into git
6. Create a new version tag for the new release and push to the repository, for example:

   ```
   git tag frontier-squid-$FSQUIDVER
   git push --tags
   ```

7. scp RELEASE_NOTES dbfrontier@frontier.cern.ch:dist/frontier-squid-releasenotes.txt
8. Create a tarball:

   ```
   mkdir /tmp/extract
   cd /tmp/extract
   wget -qO- "https://gitlab.cern.ch/frontier/tarballs/frontier-squid4/repository/archive.tar.gz?ref=frontier-squid-$FSQUIDVER" |tar xzf -
   ```

Release procedures for frontier-related software
mv * frontier-squid-$FSQUIDVER

tar czvf ../frontier-squid-$FSQUIDVER.tar.gz frontier-squid-$FSQUIDVER

cd ..

rn -rf extract

scp /tmp/frontier-squid-$FSQUIDVER.tar.gz dbfrontier@frontier.cern.ch:dist

9. Update InstallSquidTarball twiki with new version

frontier-awstats tarball

Basically similar to the frontier-squid tarball except without the copy of the release notes or twiki update. The only user is the rpm.

1. Set an environment variable for the new version, for example:

   FAWSTATSVER=6.9-5

2. Commit all changes into git

3. Create a new version tag for the new release and push to the repository, for example:

   git tag frontier-awstats-$FAWSTATSVER
   git push
   git push --tags

4. Create a tarball:

   mkdir /tmp/extract
   cd /tmp/extract
   wget -qO- "https://gitlab.cern.ch/frontier/tarballs/frontier-awstats/repository/archive.tar.gz?ref=frontier-awstats-$FAWSTATSVER"|tar xzf -
   mv * frontier-awstats-$FAWSTATSVER
   tar cvf ../frontier-awstats-$FAWSTATSVER.tgz frontier-awstats-$FAWSTATSVER
   cd ..
   rm -rf extract

5. scp /tmp/frontier-awstats-$FAWSTATSVER.tgz dbfrontier@frontier.cern.ch:dist

RPMs

Frontier rpm releases have these steps in common (replace "package" with individual package name):

1. If updating the tomcat version in frontier-tomcat, compare its conf/server.xml with src/conf/server.xml.O. If there are changes, update server.xml.O and server.xml.proto.

2. Commit your changes to git.

3. Make sure you have created a working directory according to the instructions under "Building frontier rpms" step 1, a directory called "rpms" where each repository for doc, scripts, and all frontier-* packages are checked out under their own name.

4. Update frontier-package/SPECS/frontier-package.spec with the new version, release4source, and releasenum, and commit to git.

5. If the spec file has a %changelog, add a new entry and commit to git, otherwise edit frontier-package/doc/frontier-packageRELEASE_NOTES with the new version number and release notes, and commit that to git.

6. Do git push to upload changes to the git server.

7. In directory frontier-package/scripts on an SL6-based system, run ../../scripts/rpmbuild_.sh including entering your passphrase for the signature.

8. In the same directory, run ../../scripts/rpm2web.sh /home/dbfrontier/dist/rpms-debug to publish the release to the cernvm-frontier-debug repo. You need to be on a machine on which you can do 'sudo -u dbfrontier' and which is set up with an ssh key to be able to scp from that account to dbfrontier@frontier.cern.ch.

9. In the same directory, run ../../scripts/rpm2web.sh to install to the regular cernvm-frontier repo. Answer "yes" to the prompt.

10. In the same directory, run ../../scripts/gittag.sh to create the release tag.
11. Send a release announcement to frontier-talk@cern.ch.