

Tagging Dev Builds

Overview

The idea of running an automated build over any tier was a quick and efficient method to completing a build. However, some problems arose due to these independently-tiered builds. Since these builds were performed on one tier, the specific version could only be tested on tier's environment. If this specific build was needed to be tested on a different environment, the folders could be transferred from that tier to another. Although this method is deemed acceptable, uncaredful tier-specific configuration changes, or the possibility of file transfer errors have occasionally occurred.

To eliminate these possibilities, a procedure has been created to tag a specific build created on the dev tier. A newly created build would be tagged from the build folder using the `tag.build.csh`:

Tagging a Build

`tag.build.csh`

Before proceeding, ensure that you have permission to checkout any Stanford-related projects as well as NCI projects from SVN. If you are running the script for the first time, it will prompt you for a username and password. Notify the NCICB systems group for NCI-related access, and email Tim Redmond or Tania Tudorache for Stanford-related access.

This script is available for any build, and must be run from within the build folder. It will gather all the revision numbers from the build and store all the files involved into a separate 'labeled' directory. Before the script runs, it will prompt a flag in the console indicating the current Protégé release number, as well as the tag label. The general naming convention given to the label is usually the most current build version number (ex. 1.4.0.88). If this information looks correct, the tagging can be executed. Below is a snippet of what to expect in console once '`tag.build.csh`' is executed:

```
Protégé Version: 1.4
Tagging with label: 1.4.0.105
Continue \[EVS:n\] y
-----
FYI: Created directory: [https://gforge.nci.nih.gov/svnroot/protégegui/collaboratedevterminologytools/nci-branches/tags/1.4/1.4.0.105]
Committed revision 3152.
-----
FYI: Created directory: [http://smi-protege.stanford.edu/repos/protege/nci-branches/tags/1.4/1.4.0.105]
Committed revision 13990.
-----
FYI: Created tag: [http://smi-protege.stanford.edu/repos/protege/nci-branches/tags/1.4/1.4.0.105/lucene-query]
Committed revision 13991.
```

You can see that the script will create the 'tags' directory, with the release number and tag label, and start pulling the revision numbers of each project from the original build. Once completed, the build is now ready to be reproduced in another tier.

Building a Tagged Version

Once a build has been labeled, log into the tier where you want to build using this tagged version. Once in the environment, navigate to `./app/protege/repo/bin` to initiate a new build. You will be using the familiar '`build.tier.csh`' script to kick off the build, however, a few more parameters other than the tier argument will be needed to create the tagged build:

`build.tier.csh -tier <tier> -svn-url tag -n <build number>`

The '`-tier`' parameter specifies which tier you are building in. Please refer to the [automated build procedures](#) for a list of tier arguments. The '`-svn-url`' parameter specifies what svn-url.properties the build should be referring to. This properties file contains all the specific directories of the projects that will be pulled from svn. A properties file has been created for tagged builds (**`svn-url-tag.properties`**). This file will utilize the tag label. To reference this file, simply use 'tag' as the argument for this parameter. The '`-n`' option refers to the build number for the build. For example, if the tagged build was labeled '1.4.0.95', 95 would be passed in for -n.

Once all parameters have been included, execute the build, and it will start pulling the files from the labeled directory. The build should be exactly the same as what was built in dev, except for the [tier-specific info](#).