# Protege Server Build Procedures

## Protégé Server Build Procedures

**Contents of this Page**



### Objective

This procedural guide is meant for use in performing a new Protégé server build for both the NCI Thesaurus and BiomedGT instances. Protégé server builds can be performed either manually, or via automated run scripts. Generally, using the automated scripts would be less time-consuming. However, it is recommended that going through the manual process (at least once) be performed to understand the concepts and the workflow of a Protege build. Doing so will facilitate troubleshooting any future problems during a build.

Server builds can include items such as bug fixes, software updates, and updates to the database schema. Build procedures will differ depending upon the aforementioned scenarios.

### Background

Protégé is a free, open-source editor, specific for working with ontologies.

### Component Overview

- **MySQL** - All Protégé data resides in a MySQL database.
- **Protégé Server -** The Protégé Server itself is an open-source application developed by Stanford University. It pulls data from a MySQL database and serves it up through RMI. The Protégé Server is compiled and run with Java 5.0. We currently have it running on a Linux OS.
- **Explanation Server** - The Explanation server is also an open-source application developed by the Clark & Parsia group in the Washington D.C. area. The server makes a connection to a specified Protege database, and classifies all concepts. The server will then allow users to pull up and view any concept's super-concepts as well as sub-concepts.
- **Protégé Client (Editor) -** The Protégé Client is the editing tool which the editors utilize to create, update, or view the various concept and concept property data. Data can be accessed from the database via the Protégé Server. The editors have the client running on their Windows workstations.

### Steps for Jenkins build

There are 2 parts to the Jenkins build, the compilation and the deployment.  Navigate to https://evs-jenkins.nci.nih.gov and find "default >> lower >> protege".  Select "protege-build" and select the tag given to you by the developers.  The build will take 10-15 minutes.  At the conclusion, go to "protege_deploy_dev" and deploy the tag you just built.


# Steps to complete a Manual Protégé Server Build Process for Dev, QA, and Stage Tiers

You will need tier-specific info for Dev, QA, Stage before commencing the build. Production-tier information is separated, since the information is a bit different than the other tiers.

## Backup Current Database Project

If there is an existing database project file which the Protege server is referencing, it would be a good idea to save a backup copy, in case the project ever needs to be rolled back to due to new build complications. Please follow the steps below to export a copy of the database to a file. If this is the very first build from scratch, scroll down to the 'Protege Updated File Checkout and Build' section.

- Open Reflection X to connect to your server instance (obtain username and password info from the system administrator).
- Before shutting down the server, we will need to create a backup of the project before updating. From the terminal, go to /usr/local/protégé /Protégé_x.x/<PROTÉGÉ INSTANCE>/Protégé.Client-x.x.x and run sh run_protege.bat.
  **Protege_x.x** represents the current Protégé release version number.
  **PROTÉGÉ INSTANCE** represents either BiomedGT or NCIThesaurus ontology.
  **Protégé.Client-x.x.x** represents the current build version number.
- Connect with the database project on the server. The database project should have the following naming convention: <PROTEGE INSTANCE>-YYMMDD-DB.pprj.
- Click File, Convert Project to Format, Select OWL / RDF File, Enter location and name of new file (Use '<PROTÉGÉ INSTANCE>-YYMMDD-File. owl' as the filename convention), Click OK. This file can be saved at app/protégé/data/Protégé_x.x/<PROTÉGÉ INSTANCE>. Depending on the size of the database, conversion can usually take up to 30 minutes.
- Exit the Protégé client WITHOUT saving.

## Protégé Updated File Checkout and Build

- Checkout can be performed from a Windows GUI (if the Tortoise SVN client tool is installed) or, if via command prompt on the Linux server.
- To run from the Windows GUI: Create the following folder structure: nciscn<YYMMDD><VERSION>/protegegui/. From this location, right click, select 'SVN Checkout'. A window will appear asking for the source and target of files. Protégé files are located in NCI SVN, and can be checked out from: https://ncisvn.nci.nih.gov/svn/protegegui
- To run from command prompt on the server, navigate to /app/protégé/repo, and create the following directory ncisvn<YYMMDD><VERSION> /protegegui/ and run the svn checkout:

**Note**

To checkout the Protege source code, you can use the following target:

```
svn co https://ncisvn.nci.nih.gov/svn/protegegui/collaborativedevterminologytools/scripts/trunk/
```

- There may be instances where you will need to modify a few targets within the branches tag to reflect changes in other projects. In this case, you must verify that the proper 'svn-url.properties file is being referenced in the build.xml file. Each build will hold 2 separate svn-url.properties files. One is usually the default, which builds from the trunk folder. There is also a customized file that will build from the branches folders. Example code snippets are below for each file:

```
svn-url-trunk.properties (default)

'ncitab.url=$nci.base.url/protegegui/*trunk*'
 ncitab.project=$projects/ncitab

nciconcepthistory.url=$nci.base.url/protegehistory/NCIConceptHistory/*trunk*
nciconcepthistory.project=$projects/nciprojecthistory

ncievshistory.url=$nci.base.url/protegehistory/NCIEVSHistory/*trunk*
ncievshistory.project=$projects/ncievshistory

classification.url=$nci.base.url/classification/*trunk*
classification.project=$projects/classification

*svn-url-1.4.properties (customized)*

ncitab.url=$nci.base.url/*nci-branches/1.4*/protegegui
ncitab.project=$projects/ncitab

nciconcepthistory.url=$nci.base.url/*nci-branches/1.4*/NCIConceptHistory
 nciconcepthistory.project=$projects/nciconcepthistory

ncievshistory.url=$nci.base.url*/nci-branches/1.4*/NCIEVSHistory
ncievshistory.project=$projects/ncievshistory

classification.url=$nci.base.url/*nci-branches/1.4*/classification
classification.project=$projects/classification
```

- You can either keep the newly created build folder in your local directory and build from there (assuming ant is installed locally), or transfer it to the server machine under app/protégé/repo. Utilize secure ssh client to transfer the svn checkout files over to the server.
- Navigate to <PATH>/trunk and open the 'build.properties' file and modify the 'PromptNCIPlugins.properties settings' section within the file. This info represents the tier's database connection information, and will need to be set before running the ant build commands. To edit the file using a Unix VI editor within the server console, type 'vi local.properties'. The file should now be displayed in the console. ype 'Shift + I (eye)' to enable editing, and utilize the arrow keys to navigate to the PromptNCIPlugins.properties section, and fill in the default database connection values with tier-specific values. Upon completion, type 'esc' to move out of editing mode, and type ':wq' to overwrite, save changes, and exit the file back to the console prompt.
- Within the same directory, open the 'version.info' file and modify the current version to the latest version (if necessary) following the same steps as above. By default, the version number will display as 1.1.0. Save changes following the instructions above.

## Protege Server Build Ant Commands

Run the following ant commands within the same directory as the build.xml file. The build.xml is usually stored in the same directory as the 'local. properties' and 'version.info' file. Once the build.xml file is found, the ant build targets are now ready to be run:

- Run '*ant checkout*'. This command will check out all files from Stanford University that were not included, but are essential to the Protégé Build.
- Upon successful checkout, run '*ant install*.' This process will compile all checked out files and will create a build/dist and build/archive subdirectory. The build/dist folder will contains updated Protégé client, server, admin, and explanation folders.
- Run *'ant archive*' to create zip files for each of the Protégé packages. The process will also create an ncibuilder.zip file. This file contains all files for the entire project. The file will be created at <PATH> /trunk/build/archive.
- Transfer the folders to usr/local/protégé/Protégé_x.x/<PROTÉGÉ INSTANCE> on the server machine. A copy of the client folder should also be saved on your local machine, to test a successful connection to the server after the build.

> **Note**
>
> When copying the client to any server tier from Windows, DO NOT copy the JRE folder. A copy of the JRE should already installed on the server box, and copying over an existing JRE will take too long.

## Configuring New Protege Build

If the build involves software updates that include any database code or schema changes, skip to the 'Protege Software Updates & Configuring New Database Project' section. For builds that will only require software updates connecting to an existing database project, please follow the 'Protege Software Updates' section.