

# Archive - Protege BiomedGT Admin Guide Cheatsheet v1.1

## Archive - Protégé BiomedGT Admin Guide Cheatsheet v1.1

### Contents of this Page

- [Servers](#)
- [Related Programs/Services](#)
- [Relevant Directory Structure in the various boxes/VMs](#)
  - [Production](#)
  - [Dev, QA, Stage](#)
  - [Dev, QA, Stage SVN files for builds](#)
- [Production login and basic server operations](#)
- [Prompt Accept/Reject cycle](#)
- [Database Load from File](#)
- [Configuring new project](#)
- [Configure protege server for a project](#)
- [Classification \(Standalone mode\)](#)
- [Classification \(client/server mode\)](#)

## Servers

cbiows501 - production  
cbapp-q1001 - qa  
cbvapp-d1012 - dev (VM)  
cbvapp-s1005 - stage (VM)

## Related Programs/Services

Reflection X (using "Direct" or xterm over SSH connection)  
F-Secure SSH Client  
F-Secure SSH File Transfer

Username and password advertised separately.

## Relevant Directory Structure in the various boxes/VMs

### Production

/usr/local/protege/Protege.Server-x.y.z  
/usr/local/protege/Protege.Client-x.y.z  
/usr/local/protege/Explanation.Server-x.y.z  
/app/protégé/data

### Dev, QA, Stage

/usr/local/protege/Protégé\_x.x.x/[EVS:PROTÉGÉ INSTANCE]/Protege.Server-x.y.z  
/usr/local/protege/Protégé\_x.x.x/[EVS:PROTÉGÉ INSTANCE]/Protege.Client-x.y.z  
/usr/local/protege/Protégé\_x.x.x/[EVS:PROTÉGÉ INSTANCE]/Explanation.Server-x.y.z

### Dev, QA, Stage SVN files for builds

/app/protege/repo/<protégé release folder>

## Production login and basic server operations

If the protege server is brought down for maintenance, classification, Prompt analysis, and so on, the RMI can continue running. When all the services need to be started (e.g. after a reboot), follow this order (see bullets below for details):

- start rmi
- start Explanation server
- start Protege server

### Remote login

- Use reflection X (or SSH client if running the Protege client is not required)

### Start rmiregistry (configure script to use port 2633 in PROD)

- From '/usr/local/protége/Protege.Server-x.y.z'
- Run './run\_rmiregistry.sh'

### Shutdown rmiregistry

- Kill from OS (identify with `ps -ef | grep rmiregistry`)

### Start Explanation server

- From '/usr/local/protége/Explanation.Server-x.y.z'
- Run with option to periodically save the reasoner state at a desired time frame: `./start_explanation_server.sh --port 8090 --protege-standalone /path/to/database/project/ --persistence <numeric time value>`
- Run w/out checking for any changes in ontology: `./start_explanation_server.sh --port 8090 --protege-standalone /path/to/database/project/ --no-ontology-sync`

### Re start Explanation server

- From '/usr/local/protége/Explanation.Server-x.y.z'
- Run with option to restore to previous saved state of reasoner, as well as to check for any recent changes in the ontology from the previous saved state: `./restart_explanation_server.sh --restore --persistence`

### Verifying Explanation servers is starting up correctly

- From '/usr/local/protége/Explanation.Server-x.y.z'
- Run `tail -f nohup.out`

### Shutdown Explanation server

- From '/usr/local/protége/Explanation.Server-x.y.z'
- Run `./stop_explanation_server.sh`
- Note, after a shutdown, the server can't be "restarted", it must be "started" instead.

### Start Protege server

- From '/usr/local/protége/Protege.Server-x.y.z'
- Run `./run_protege_server_nci.sh`

### Shutdown Protege server

- From '/usr/local/protége/Protege.Server-x.y.z'
- Run `./shutdown_protege_server.sh localhost: 2633`

### Verifying Protege servers is starting up correctly

- From '/usr/local/protége/Protege.Server-x.y.z'
- Run `tail -f console.txt`

### Start Protege client

- From '/usr/local/protége/Protege.Client-x.y.z'
- Run `./run_protege_BGT.sh`

## Prompt Accept/Reject cycle

- Advertise (email/chat) the server is going to be shut down
- Clean up Pre-Merged, and Pre-Refined concepts.
- Remote login to the host computer using Reflection X
- Shut down Protege server
- Shut down Explanation server.
- Start Protege client.
- Connect to database project by following the path: `/app/protége/data/BiomedGT.pprj`.
- Enable PromptTab in menu (in Project->Configuration dialog).
- Compare against previous baseline file/database project (pprj file).
- Accept/reject changes.
- Save concept\_history (in "save" icon in Table View).
- Disable PromptTab (in Configuration dialog).
- Adjust baseline version in Metadata tab.
- In the Protégé GUI menu, click 'Change' , 'Clean up Changes Ontology'. Ensure that 'Delete Changes' and 'Archive Changes' remain checked. Click 'OK'. After the process has been completed, click 'Change' , 'Save Changes Ontology'
- Export file for next Prompt comparison or publication.
  - Convert Project to Format -> OWL / RDF Files.
    - This creates a new project .pprj and associated OWL file, give it a name with a date suffix, example BiomedGTFile-071128.pprj.
- Close project without saving, and exit the client.
- In the Reflection X console, navigate to `/app/protége/data/` and look for any BiomedGT accept files: (BiomedGT-accept.pprj, .pjrn, .rdf, .rdfs) and remove them from the directory by typing `'rm BiomedGT-accept.pprj'` and so on for the other files.
- Start Explanation server (always restart explanation server first before the Protégé Server).

- For Production tier at the end of the month:
  - i) Reopen Protégé client.
  - ii) Click 'Project', 'Configure', enable 'OWLClasses Tab'.
  - iii) Click 'Reasoning', 'Classify Taxonomy'.
  - iv) Save the Inferred Ontology using the save button in the Inferred Hierarchy section. Give the naming convention 'BiomedGT-Inferred-YYMMDD.owl'\*
- Startup Protege server.
- Verify Protege server is up.
- Verify Explanation tab communicates with Explanation server.
- Exit remote program.
- Advertise that the server is back up.
- For end-of-the-month publications, after performing a prompt, notify Tracy, John, or Rob of the availability of the new baseline.
- In the notification include: i) the filename of the baseline and (ii) the length of time for the Prompt/export to file to complete.
- The raw exports (OWL files only) will be archived in the private FTP site, and the processed exports in the anonymous FTP site. This will help keep the Protégé server clean and organized.

## Database Load from File

- Remote login to the host computer using Reflection X.
- cd to /usr/local/protege/Protege.Client-x.y.z.
- Start protege ("./run\_protege\_BGT.sh").
- Load file from /app/protege/data/(new project).
- Convert project to format (Clark & Parsia owl / rdf database).
  - initial project BiomedGT.pprj (as opposed to baselines).
- Configure project (see below, include Save but don't Exit).
- Convert project to format (owl / rdf file) for a file baseline, with date.
- Exit Protege without saving (the Save gets done automatically during the conversion).

## Configuring new project

- (already logged in in the server box).
- Enable tabs:
  - nciedittab
  - advanced query
  - properties
  - metadata
  - changes
  - explanation tab
- Disable extra tabs (OWLClasses, OWLInstances, OWLForms).
- Enable journaling.
- Disable redo/undo.
- Set OWL Preferences items:
- Configure URL to the correct explanation server.
- Set preferences to read Manchester Syntax.
- Disable drag and drop in owl preferences.
- Disable test of deprecated classes/properties.
- Set Visibility to uncheck all MetaData Items.
- Set Reasoning menu items:
- Click on Reasoning menu, and ensure that Clark & Parsia Custom Protégé 3.x Reasoner is selected on the menu list.
- Click on Reasoning menu, C & P Custom Protégé 3.x Reasoner Details, and ensure the 'Synchronize Changes with Reasoner' box is checked, and click OK.
- Set transitivity.
- Set display slots to read rdfs:label for NCI Edit tab as well as Property Tag.
- Generate Lucene indices from Lucene menu. Ensure both Phonetic and Standard boxes are checked prior to indexing.
- Start explanation server, point to the newly created project. Once the server's up and running, save the project.
- Exit protege.

## Configure protege server for a project

- Edit run\_protege\_server\_nci.sh.

a) to include the proper directory path for the nci\_metaproject file - /app/protege/data/metaproject/.

b) to use the correct rmi port (use 2633).

## Classification (Standalone mode)

- Advertise the protege server is going to be shutdown.
- Remote login to the host computer using Reflection X.
  - Username and password are advertised separately.
- Shutdown Protege server.
- Restart Explanation server.
- Start protege client.
- Connect to database project.
  - in /app/protege/data/.
  - file - BiomedGT.pprj.

- Enable OWLClassesTab (results are shown there).
- From the OWL menu, select "Classify taxonomy", it takes about 20 minutes.
- Once classification is complete, an inferred hierarchy panel and a list panel are displayed.
- Accept retreeing suggestions in the list panel as per guideline.
- Generate table to pass around to editors and mail it.
- Process table from previous classification (accept/reject).
- Close project without saving.
- Start protege server.
- Verify protege server is running.
  - connect to the server from the client.
  - exit the client.
- Exit terminal.
- Exit reflection X.

## Classification (client/server mode)

- From the GUI, enable OWLClassesTab (results are shown there).
- From the OWL menu, select "Classify taxonomy", it takes about 20 minutes.
- Once classifications is complete, an inferred hierarchy panel and a list panel are displayed.
- Accept retreeing suggestions in the list panel as per guideline.
- Generate table to pass around to editors and mail it.
- Process table from previous classification (accept/reject).
- Close project without saving.
- Exit the client.