

# Protege NCI Thesaurus Admin Guide Cheatsheet v1.0

## Protégé NCI Thesaurus Admin Guide Cheatsheet v1.0

### Contents of this Page

- [Servers](#)
- [Related Programs/Services](#)
- [Related Directory Structure in the Various Boxes/VMs](#)
  - [Production](#)
  - [Dev, QA, Stage](#)
  - [Dev, QA, Stage File Directory for Builds](#)
  - [Build Notes](#)
- [Production Login and Basic Server Operations](#)
  - [Prompt Accept/Reject Cycle](#)
  - [Database Load from File](#)
  - [Database reload into an existing project](#)
- [Configuring a New Project](#)
- [Configuring the Metaproject](#)
- [Configure Protégé Server for a Project](#)
  - [Classification \(Standalone Mode\)](#)
  - [Classification \(Client/Server\) Mode](#)
- [Troubleshooting](#)

## Servers

| Name         | Description                 |
|--------------|-----------------------------|
| cbiows501    | Production (protegeqa user) |
| cbvapp-s1005 | Stage (protegeadm user)     |
| cbapp-q1001  | QA (protegeadm user)        |
| cbvapp-d1012 | Dev (protegeadm user)       |

## Related Programs/Services

- MobaXterm Professional
- Username and password advertised separately

## Related Directory Structure in the Various Boxes/VMs

### Production

- `/usr/local/protegeqa/NCIThesaurusGroup/Protege.Server-x.y.z`
- `/usr/local/protegeqa/NCIThesaurusGroup/Protege.Client-x.y.z`
- `/usr/local/protegeqa/NCIThesaurusGroup/Explanation.Server-x.y.z`
- `/app/protegeqa/data/[EVS:PROTEGE INSTANCE]`

### Dev, QA, Stage

- `usr/local/protege/Protege_x.x.x/[EVS:PROTEGE INSTANCE]/Protege.Server-x.y.z`
- `usr/local/protege/Protege_x.x.x/[EVS:PROTEGE INSTANCE]/Protege.Client-x.y.z`
- `usr/local/protege/Protege_x.x.x/[EVS:PROTEGE INSTANCE]/Explanation.Server-x.y.z`

### Dev, QA, Stage File Directory for Builds

- `/app/protege/repo`

### Build Notes

- The automated build script will prompt for various configuration items (for files such as the run scripts, PromptNCIPlugin.properties, etc).
- Do not use the default localhost for any IP addresses. Always use the machine name, and include the port.

- Do not include PromptNCIPlugin.properties with a distributed client.
- Set the max code + 1 in codegen.dat

## Production Login and Basic Server Operations

- If the Protégé server is brought down for maintenance, classification, Prompt analysis, and so on, the RMI can continue running. While all the services need to be started (e.g. after a reboot), follow this order (see the following bullets for details):
  - start RMI
  - start explanation server
  - start Protégé server
- Remote login
  - Use MobaXterm (or SSH client if running the Protégé client is not required).
- Start RMI registry (configure script to use port 2822 in production)
  - From /usr/local/protégeqa/NCIThesaurusGroup/Protege.Server-x.y.z
  - Run ./run\_rmiregistry.sh
- Shutdown rmiregistry
  - Kill from OS (identify with ps -ef | grep rmiregistry)
  - Run kill -9 [rmi pid]
- Start explanation server
  - From /usr/local/protégeqa/NCIThesaurusGroup/Explanation.Server-x.y.z
  - Run with option to periodically save the reasoner state at a desired time frame: ./start\_explanation\_server.sh --port 8092 --protege-standalone /path/to/database/project/ --persistence <numeric time value>
  - Run w/out checking for any changes in ontology: ./start\_explanation\_server.sh --port 8092 --protege-standalone /path/to/database/project/ --no-ontology-sync
- Restart explanation server
  - From /usr/local/protégeqa/NCIThesaurusGroup/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 --persistaece
- Verifying explanation servers is starting up correctly
  - From /usr/local/protégeqa/NCIThesaurusGroup/Explanation.Server-x.y.z
  - Run tail -f nohup.out
- Shutdown explanation sever
  - From /usr/local/protégeqa/NCIThesaurusGroup/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 Protégé Server
  - From /usr/local/protégeqa/NCIThesaurusGroup/Protege.Server-x.y.z
  - Run ./run\_protege\_server\_nci.sh
- Shutdown Protégé Server
  - From /usr/local/protégeqa/NCIThesaurusGroup/Protege.Server-x.y.z
  - Run ./shutdown\_protege\_server.sh localhost:2822
- Verifying Protégé servers is starting up correctly
  - From /usr/local/protégeqa/NCIThesaurusGroup/Protege.Server-x.y.z
  - Run tail -f console.txt
- Start Protégé client
  - From /usr/local/protégeqa/NCIThesaurusGroup/Protege.Client-x.y.z
  - Run ./run\_protege\_NCIT.sh

## Prompt Accept/Reject Cycle

- Advertise (email/chat) the server is going to be shutdown
- Clean up Pre-Merged, and Pre-Retired concepts.
- Remote login to the host computer using Reflection X
- Shut down Protégé server
- Shut down explanation server
- Start Protégé client
- Connect to database project by following the path: /app/protégeqa/data/Protege\_1.2.3/NCIThesaurus/Thesaurus.pprj
- Enable Prompt Tab 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' tab)
- 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 the next Prompt comparison or publication
  - Click 'File' -, 'Convert Project to Format', 'OWL/RDF Files'

- this creates a new project .pprj file and associated OWL file, give it a name with a date suffix, example: *ThesaurusInterim-071120.pprj*
- Close project without saving, and exit client
- In Reflection X console, navigate to */app/protegeqa/data/Protege\_x.x.x/NCIThesaurus* and look for any Thesaurus accept files: (Thesaurus-accept.pprj, pjrn, .rdf, rdfs) and remove them from the directory by typing *'rm Thesaurus-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:
  - Reopen Protégé client
  - Click 'Project', 'Configure', enable 'OWL Classes Tab'
  - Click 'OWL', 'Classify Taxonomy'
  - Save the Inferred Ontology using the save button on the Inferred Hierarchy section. Give the naming convention 'Thesaurus-Inferred-YYMMDD.owl'
- Start up Protégé server
- Verify Protégé server is up
- Verify explanation tab communicates with the 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: (a) the filename of the baseline and (b) the length of the 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 keep the Protégé server clean and organized.

## Database Load from File

- Remote login to the host computer using Reflection X
- cd to */usr/local/protegeqa/NCIThesaurusGroup/Protege.Client-x.y.z*
- Start Protégé client (*'./run\_protege\_NCIT.sh'*)
- Load file from */app/protegeqa/data/Protege\_x.y.z/NCIThesaurus (new project)*
- Convert project to format (OWL/RDF database)
  - initial project - Thesaurus.pprj (as opposed to baselines)
- Configure project (see below, include Save, but don't Exit).
- Exit Protégé without saving (the Save gets done automatically during the conversion)

## Database reload into an existing project

Preconditions – Thesaurus running normally

1. Thesaurus tables exist
2. ChaO tables exist
3. Metaproject exists
4. Prompt.properties exists and configured
5. Ncitab.xml exists and configured
6. Shell scripts all exist and configured
7. Prompt complete (if needed)
8. File exported

Steps

1. Stop protégé server.
2. Stop explanation server.
3. Start Protégé, New Project->Check Create from Existing Sources, select OWL/RDF Files->Find file->OWL DL->Finish
4. Save to temp project Thesaurus\_Temp.pprj
5. Change to database project File->Convert Project to Format-> Select **Clark-Parsia Protégé OWL/RDF Database**->Enter the JDBC URL using the database ->Table (Thesaurus)-> Username (\*\*\*\*\*)->Password(\*\*\*\*\*)
- a. Overwrite existing database tables.
6. File->Save Project to the temp project
7. Close client. Reopen Thesaurus.pprj
  - a. Check that everything is configured correctly and the database can be reached.
8. Close client
9. Start explanation server and monitor
10. Start protégé server
11. Start client pointed at server
12. Reindex
13. Verify it works and returns queries
14. Classify

## Configuring a New Project

- (already logged-in in the server box)
- Enable tabs in this order

- NCIEditTab
  - LuceneQueryPlugin
  - OWLProperties
  - OWLMetadataTab
  - ChangesTab
  - ExplanationTab
- Upon enabling the ChangesTab, click OK
  - Dialogue will pop up. Select 2nd option 'Create ChAO stored in database' and click enable change tracking. Click OK.
  - Create new DB dialogue table gets "annotation\_" prefixed
  - Enter DB credentials
  - Click OK
- File->Save Project
- Configure->Deselect Changes tab, Options tab->Track changes should be checked
- Disable other extra tabs (OWLClasses, OWLInstances, OWLForms)
- Set display slots to read 'Preferred Name' for the NCI Edit tab as well as Property Tag within the Forms Tab for:
  - owl:Class
  - DatatypeProperty
  - code
  - ObjectProperty
- Configure URL for explanation service.
- Goto OWL->Preferences
  - Disable Drag and Drop, verify reasoner URL is correct
  - Visibility->uncheck rdfs:Class, owl:Class, rdf:Property
  - Datatypes->rdf:XMLLiteral should be checked
  - Tests->uncheck 'Find TODO list items' and 'List deprecated classes and properties'
- Reasoning->Select Clark & Parsia
- Reasoner Details->Click Synchronize Changes with Reasoner, Reasoner Behavior select Synchronize on query
- Set OWL Preferences to read 'Manchester Syntax'
- Set transitivity
- Generate Lucene indices
  - Lucene->Index Ontologies, accept all slots, Use Phonetic and Standard indexers, click OK (this should take less than 20 minutes in a Production quality environment)
  - Move lucene index from `/app/protegeqa/data/Protege_x.y.z/NCIThesaurus` to the Protege server directory
- Save project
- Exit Protégé

## Configuring the Metaproject

- Open the metaproject
- Instance tab->Click User class
  - In the Instance Browser copy an editor (to the number of users needed)
  - Enter full name
  - Change password
- PolicyControlledObject->ProjectClass
  - Verify Thesaurus instance Location is edited to the pprj location
  - Verify Annotation Thesaurus instance Location is edited to the annotation\_Thesaurus... location
    - It should be granted the 'GrpOp(World)' instance that does not contain the 'DisplayInList' instance to prevent it from showing as a selectable project on login
    - See '[DisplayInList Project Setup Instructions](#)' for more details.
  - File->Save Project->Exit

## Configure Protégé Server for a Project

- Edit `run_protege_server_nci.sh`
  - to include the proper directory path for the nci\_metaproject file - `/app/protegeqa/data/NCIThesaurusMetaProject/`
  - to use the correct rmi port (use 2822)

## Classification (Standalone Mode)

- Advertise the Protégé server is going to be shutdown
- Remote login to the host computer using Reflection X
  - username and password are advertised separately
- Shutdown Protégé server
- Restart explanation server
- Start Protégé client
- Connect to the database project
  - in `/app/protegeqa/data/Protege_x.y.z/NCIThesaurus`
  - file `Thesaurus.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 Protégé server

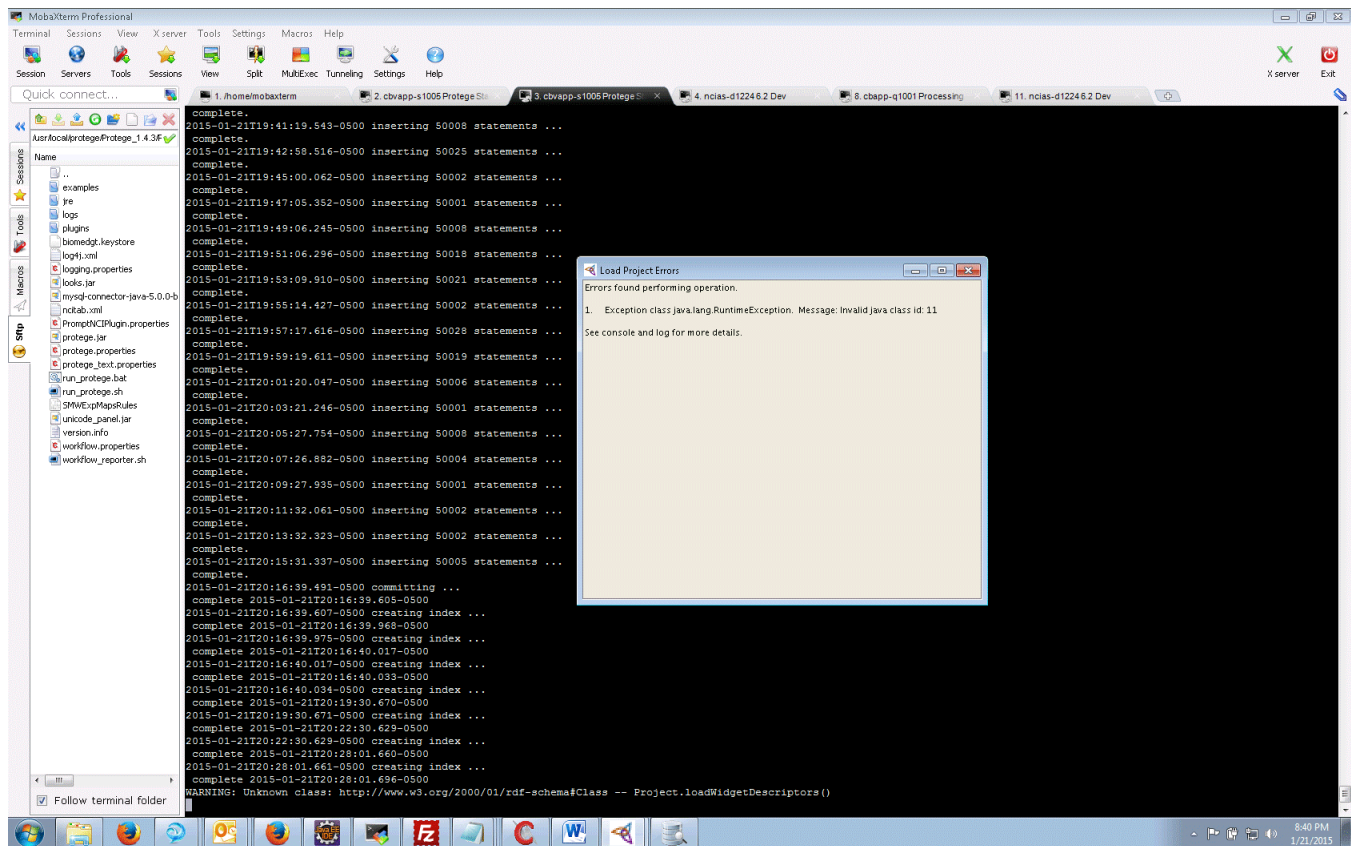
- Verify Protégé server is running
  - connect to server from the client
  - exit the client
- Exit console
- Exit Reflection X

## Classification (Client/Server) Mode

- From the client GUI, 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
- Exit the client.

## Troubleshooting

Q: After building the database, why am I getting this weird warning and Java RuntimeException?



A: You accidentally selected the 'Clark & Parsia Database' option. Start over and select the 'Clark & Parsia OWL/RDF Database' instead.

Q: I exported an owl file from my project, but now the project loads from a file. What's up?

A: When exporting an owl file, rename the project and not the owl file by itself. When renaming the project, your owl export will follow the same naming convention used.

Q: I've started a new project and am trying to enable the Changes tab, but am getting a red box and error similar to 'ncit.annotation\_thesaurus doesn't exist'

A: This is due to a hard-coding where the application will attempt to find a project file named **annotation\_PROJECTNAME.pprj**. Rename this project file if it exists so Protege can create a new one and is no longer trying to reference the old file.