

Protege Software Updates and Configuring New Database Project

Protégé Software Updates and Configuring New Database Project

A new project file will need to be created using the raw OWL file located in `/app/protégé/data/Protégé_x.x/<PROTÉGÉ INSTANCE>`. Usually newer, more fresher data is available by the editors after performing a PROMPT of production data. Each baseline file that is created after each session can be used as test data. If however, for some reason, an OWL file must be created manually, instructions are available on how to [export data from the production database](#).

- As long as no changes have been made to the behavior of the startup scripts, the following files and scripts should be copied over from the old Protégé.Server version to the new folder:

```
run_rmiregistry.sh
run_protege\_server_nci.sh
shutdown_protege_server.sh
codegen.properties
codegen.dat
```

- If you do need to use new scripts, make sure to adjust the memory upwards from the default in the `run_protege_server_nci.sh` script. The following [table](#) displays how much memory to allocate for each script, per tier. You will also need to adjust the `rmi.registry` and `rmi.server` ports, the `hostname` and the `metaproject` location in this same file. Please view Appendix A for port information for each tier.
- As long as no changes have been made to the behavior of the startup scripts, navigate to the Protégé.Client folder and transfer the following script:

```
run_protege_BGT.sh (BiomedGT) or run_protege_NCIT.sh (NCIThesaurus)
```

- If you do need to use a new script, make sure the `-Xmx` memory is adjusted upwards from the default from the [table](#).
- As long as no changes have been made to the behavior of the startup scripts, navigate to the Explanation.Server directory and transfer the following files:

```
restart_explanation_server.sh
stop_explanation_server.sh
start_explanation_server.sh
```

The 'protege_install' path will need to be modified in order to specify the correct Protégé server version.

- To edit, type '`vi _start_explanation_server.sh`' to view the script from the vi editor. Type '`shift + I`' and edit the server number to the correct version number. Press '`esc`', then '`:wq`' to save changes and exit the editor.
- If you do need to use a new script, utilize the [table](#) for the explanation server startup script. There should be no other edits that need to be made.
- Navigate to the Protégé server, and open the 'codegen.properties' file using a vi editor. To do this, type '`vi codegen.properties`'. Once the file is viewable from vi, type '`i`' to edit. The `code.prefix` value can be adjusted, depending on which Protégé Instance is being used. BiomedGT used a prefix of 'B', while NCI Thesaurus uses 'C'. Keep the seed number at 100000. Save and exit the file by typing '`esc`', then '`:wq`'.
- Open the 'codegen.dat' file in the vi editor. You will notice a number which represents a code that will be assigned to the concept code created in the database project. Set this number to 102000. Save and exit the file.
- Open the Protégé client folder, and click on '`run_protege<PROTEGE-INSTANCE>`' to open the client. Once it is open, click on "Create New Project". Select OWL/RDF File, and check "Create from existing source". Click Next.
- A window will open requesting the address of the source of the OWL file. Navigate to `/app/protege/data/Protege_x.x/<PROTEGE-INSTANCE>/` and select the OWL file, and click next. Leave the next two windows to their default values, then click Finish. Loading the file to the client will take approximately 5-10 minutes.
- *The following few steps reference the deprecated BiomedGT project and may be ignored.*
- Once the file is loaded, the ontology repository settings will need to be changed if configuring a BiomedGT project. From the menu bar, click 'OWL', and 'Ontology Repositories'. The Repository Manager window is now displayed with a list of default project repository locations underneath the 'Project repository' tab. Remove the following URL repositories:

```
http://ncicb.nih.gov/xml/owl/EVS/BGTCommonWords.owl
http://ncicb.nih.gov/xml/owl/EVS/BGTThesaurusNodes.owl
http://ncicb.nih.gov/xml/owl/EVS/External.owl
```

- There should be two URLs remaining, one referencing BFO, and another referencing protege-dc. Those will remain as is.
- The removed repositories will then be re-added to the list, but referenced locally. Accessing local copies of an external ontology increases performance as well as dependability of the ontology (the ontology URL might be down). To read locally, click on the white plus button located at the top right of the window.
- Follow the wizard by specifying the location of your local folder. Ensure that the 'Force Read-Only' box is unchecked.
- Once the window is closed, a pop-up window is displayed prompting you to save the settings and to reload the data into the client. Select the 'Reload' button, and the client should save the settings and reload the client. This step should anywhere from 10-20 minutes, depending on your machine's resources.
- *End of steps pertaining to BiomedGT*
- Once loaded (the GUI should be visible with the OWL file displayed), the project file can now be converted to a Clark & Parsia OWL/RDF database. Click File, Convert Project to Format, Clark & Parsia OWL / RDF database.
- In the database info window, enter all requested database info and click finish. Give the Project name the following naming convention: *<PROTEGE INSTANCE>-YYMMDD-DB.pprj*. Note that the JDBC URL must be in the following format: *jdbc:mysql://<databasehostname>:<port>/<database name>* (Please look at the [database info](#) for each instance and tier). Give the table name the following naming convention: *<PROTEGE-INSTANCE>_<YYMMDD>_<tier>_<release number>*. Conversion can take anywhere up to 2.5 hours.
- After conversion of the project file, you can verify that the database has been created by utilizing a DBMS (such as DB Visualizer) to view if the table has been created. Once verified, the project is now ready to be configured from the client.
- Set transitivity by clicking the properties tag, and perform a search for the 'Anatomic_Structure_Has_Location' property. Select, and check the box marked transitive. Repeat the same step for the 'Anatomic_Structure_Is_Physical_Part_Of' property.
- Set the display slots by clicking the NCIEdit tab, and clicking the "F" tab on the very far top right section of the page. You will now be within the Form Editor section. Set the 'Display Slot' drop down to 'rdfs:label'.
- Next, click the Properties tab, and click the "F" tab on the very right. Click 'OK' for the forms dialogue box. On the left hand menu, click on 'owl:DatatypeProperty', and set its display slot drop down to 'rdfs:label'. Next, click on the owl:ObjectProperty on the left hand menu, and set its display slot drop down field to 'rdfs:label' as well.
- Click 'OWL' on the menu tab, and select 'preferences'. On the General tab, Disable the drag and drop feature, and set the explanation server URL to 'http://<Protege Server Address>:<port>/explain'. The correct port per tier can be determined from the [tier-specific info page](#).
- In the Tests tab, disable test of deprecated classes/properties. This should be the fifth entry down.
- Click on the visibility tab from the window, and disable all the boxes in the MetaClasses section. Close the window.
- Click 'EditTab' on the menu bar, and select 'preferences'. Click on the 'panels' tab, and ensure that all four checkboxes are checked (Retire, Split, Merge, and Copy). Click the close button.
- Click 'Project' on the menu bar, and select 'configure'. On the Tab Widgets tab, enable only the following boxes from the Widgets tab and put in the following order:

```
NCIEditTab
LuceneQueryPlugin
OWLPropertiesTab
OWLMetadataTag
ChangesTab
ExplanationTab
```

- Click on the 'options' tab and enable journaling and disable redo/undo.
- After closing the configurations window, a changes ontology window will appear prompting you to select how to configure the annotations project. Currently, all annotations and changes are being stored into a database table. Select the radio option to create a changes ontology to a database table (Should be the 2nd button).
- A window will appear requesting all database information, Keep all fields to their default values, except the table name. Another name will need to be assigned to distinguish the annotations table. It is good practice to prefix the name of this table "annotation_[tablename]". Once the table is renamed, click 'OK'. This process should take a few seconds.
- From the menu bar, select 'Lucene' and select 'Index Ontologies'. A window will pop up. Accept all slots. Check both the 'Phonetic Indexer' and the 'Standard Indexer', and select 'OK'. After indexing has completed (this should take less than 20 minutes on a Production quality machine), a new Lucene folder will be stored in the same directory as your database project file.
- At this point, if a previous version of the Protege and Explanation servers are running, they will need to be shutdown. For best practices, the RMI registry will need to be shutdown as well. Advertise via email that the tier's current Protege instance will be brought down. To shutdown, open another SSH connection to the same server. Type 'ps -ef | grep server' to see view the current protege processes running. If both explanation and Protege servers are running, shutdown the explanation server first.
- Type '`cd /usr/local/protege/Protege_x.x/<PROTEGE INSTANCE>Explanation.Server-x.x.x. Type ./stop_explanation_server.sh http://localhost:<port>/explain'. After the executing, the following message should be displayed in the console: '_Request sent to server.' This message indicates that the explanation server has successfully communicated with the Protege server that it will be shutting down. Next, do a 'ps -ef | grep server' to see if the explanation server was killed.`

- Navigate to the previous version Protege server folder by typing '`cd ../Protege.Server-x.x.x`'. Once in the directory, type '`./shutdown_protege_server.sh localhost:<server port>`'. A message will appear on the console that the server has been shut down. The RMI registry is also shutdown from this directory. Unfortunately, there is no script to shutdown the RMI registry, however, can be killed manually. To determine which port you want to kill, refer to the [tier-specific info page](#). Once the correct port has been determined, type '`ps -ef | grep rmi`' to figure out which process ID is running the port that needs to be killed. To kill the process, type '`kill -9 <RMI Process ID>`'. Once executed, type '`ps -ef | grep rmi`' again to see if the process has been terminated.
- Now that the all servers from the previous version have been shutdown, the explanation servers for the latest version must now be restarted. Navigate to the latest version's explanation server folder by typing '`cd /usr/local/protege.Protege_x.x.x/<PROTEGE INSTANCE>/Explanation.Server-x.x.x`'.
- Once in the directory, type '`./start_explanation_server.sh --protege-standalone --port <port number> /app/protege/data/Protege_x.x.x/<PROTEGE INSTANCE>/<DB PROJECT NAME>.pprj`'. After the server has initiated start up, type '`tail -f nohup.out`' to view start up status. Within the nohup out file, the explanation server has started when you see "Jena, Classification, and Extracting" processes all complete, and a "Server started, listening on port xxxx" message.
- Once the server is up and running, go back to the client GUI, click on the 'Reasoning' menu, and ensure that "Clark&Parsia Custom Protege 3.x Reasoner" is selected on the menu list. Click on the 'Reasoning' menu again, and scroll down to "Clark&Parsia Custom Protege 3.x Reasoner Details" and ensure that the 'Synchronize on query' box is checked. Click 'OK'.
- The Changes tab will now be disabled from the GUI, since all changes will be recorded to the database. To disable, click on 'Projects', 'Configure' and uncheck the 'ChangesTab' box. Click 'OK', and the changes tab should now be gone from the GUI.
- Click File, Save Project.
- Select File, Convert Project to Format. A window will appear prompting to select a conversion file type. Select 'OWL/RDF', and store in `app/protégé/data/Protégé_x.x.x/<PROTÉGÉ INSTANCE>`. Label the file with the current date and "file". Example: BiomedGT-yyymmdd-file.pprj. Set the Language drop-down field to 'XML/RDF ABREV'. You will notice a small dialogue box displaying the conversion progress. This step may take approximately 30 minutes depending on the size of the database project. Once completed, the dialogue box should disappear, and the client should display the new file project.
- Close the project WITHOUT saving the file project (the file project automatically gets saved), however, leave the Protégé client open.
- Open the `nci_metaproject.pprj` located at `app/protégé/data/<PROTÉGÉ INSTANCE>/meta/`. Click on the project tab underneath the SYSTEM-CLASS tab on the left-hand side menu. Select the 'Instances' tab at the top of the page. You will notice an 'Instance Browser' section, and an 'Instance Editor' section. The instance browser specifies the display name of your project, and the Instance Editor specifies the path of your database project. Click on the display project name and set the correct path to your project files. There should be a project instance for both the main ontology and annotation ontology.
- Within the `nci_metaproject`, the changes ontology projects (annotation files) should also not be made visible to users when trying to access a project on the server. If this has not already been configured, please view the [DisplayInList setup instructions](#).
- To add users, select the 'User' class and within the Instance Browser, clone an existing user and modify the username and password.
- Save the metaproject and exit the client.
- Verify that the port numbers in the `run_protege_server_nci.sh` script correspond to the port numbers in `run_rmiregistry.sh`. Please refer to Appendix A for tier specific port numbers.
- Restart the RMI registry by running `run_rmiregistry.sh`. Type '`ps -ef | grep rmi`' to see if the rmi process has started.
- Restart the Protégé Server, using the `run_protege_server_nci.sh` script. The console will indicate that the server has started and is ready to accept connections.
- Connect to the server via Protégé client to confirm that the project is viewable from the `GUIrun_protege_<PROTÉGÉ INSTANCE>.sh`. Enter proper login information.
- Select the desired project from the project display window.
- The client should now display the database project file, accessed via the Protege server.
- Once tested, the client can now be [posted up on the CForge website](#) posted to the sFTP to be downloaded by users.