Installing Oracle Protégé Plug-in

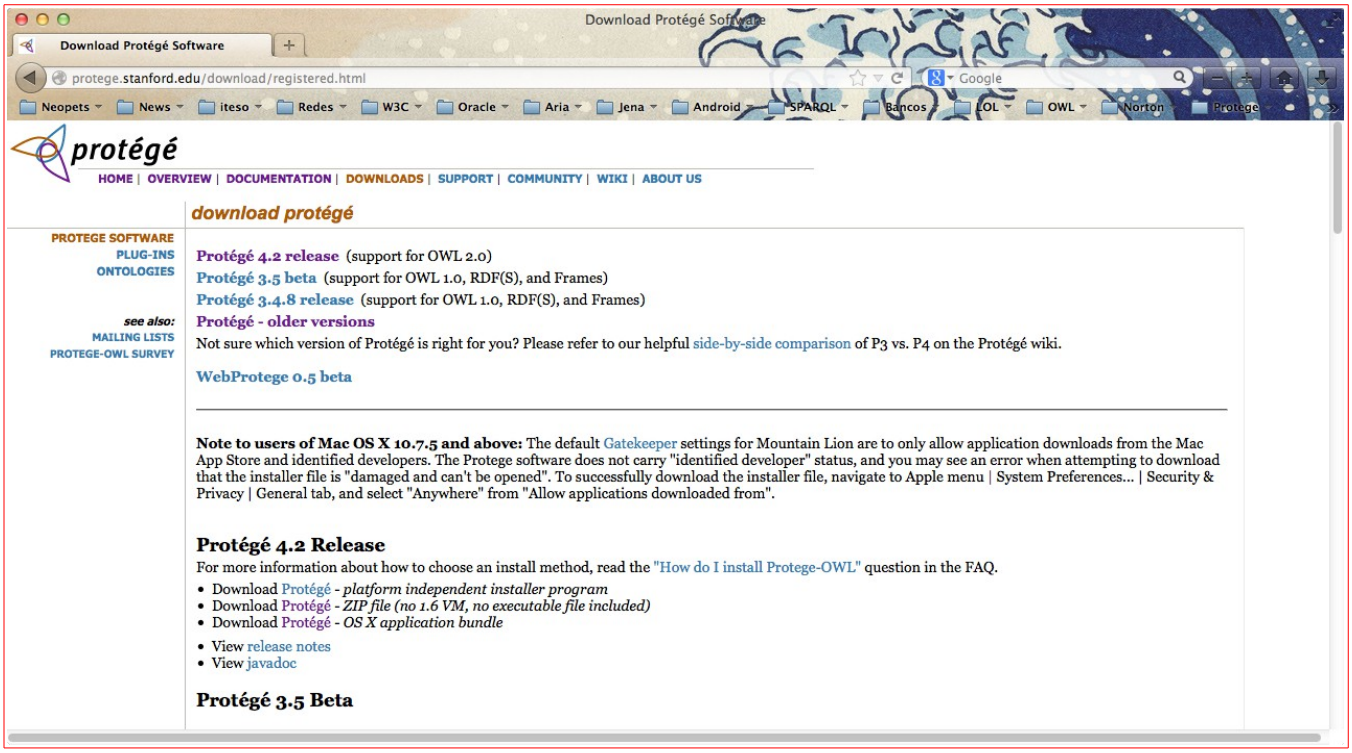
www.oracle.com
Oracle Spatial and Graph

Objective

This document describes how to install *Protégé 4.1*, enable the *Oracle Plugin* and use the plug-in.

Downloading Protégé

As a prerequisite you need to have Protégé 4.1 installed, which you can download from:
http://protege.stanford.edu/download/registered.html
You will need Protégé 4.1 and a JRE.
Protégé 4.1 is located on the "Protégé – older versions" section.



Protégé download page

Installing Protégé

When installing, there are two important locations to remember that Protégé installer will ask you:

1. Location in the hard drive to install Protégé
2. Location of the *Java Runtime Environment* you wish to use

Due to the broad support for different operating systems, it is recommended that you follow the instructions for installing Protégé in your desired OS, which are located below the download links on the download page for *Protégé 4.1*.
It is strongly recommended that you use JRE 1.6 instead of the provided one.

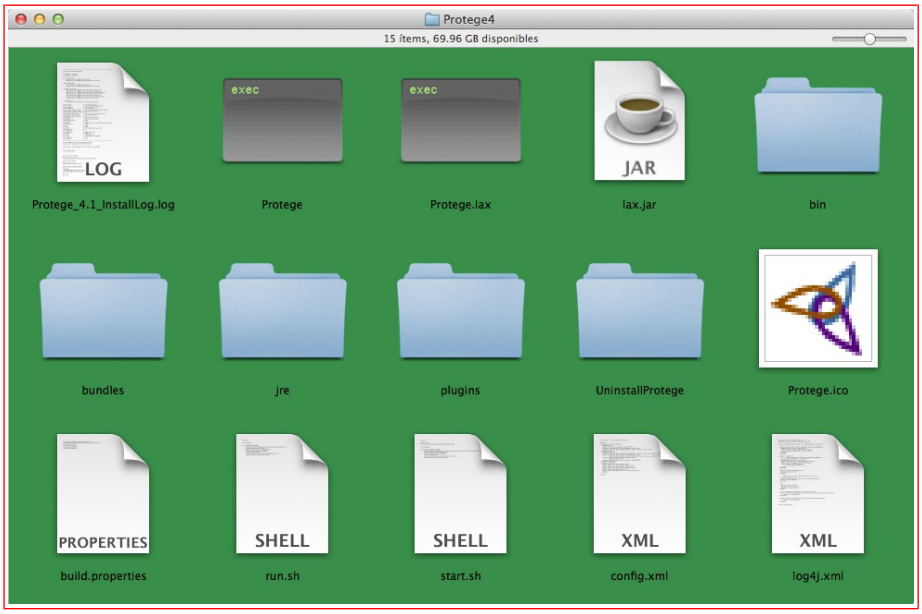
Enabling the Oracle Plug-in

Downloading the JAR

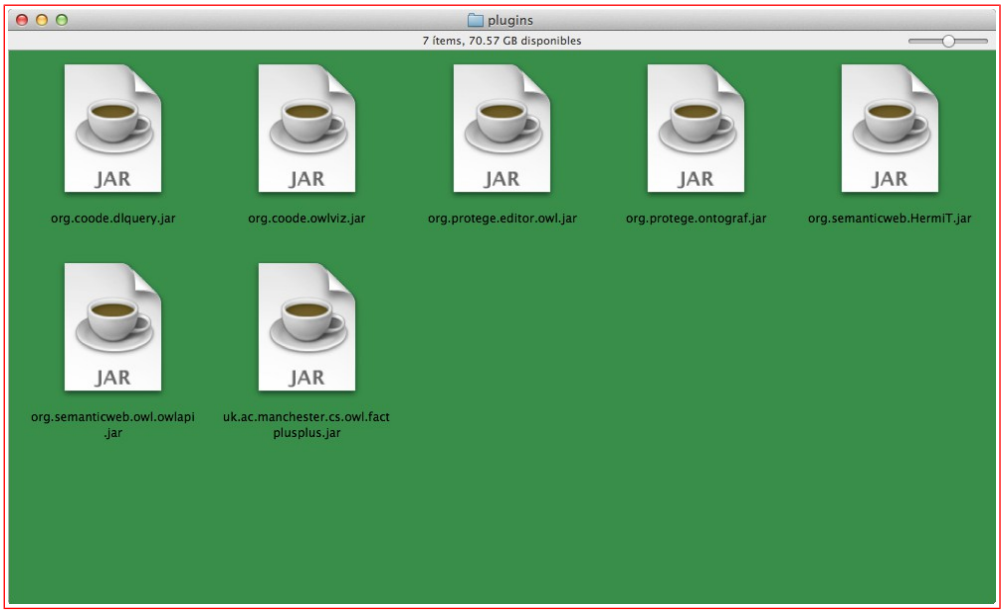
The plug-in can be downloaded from Oracle's website:
http://www.oracle.com

Adding the JAR to Protégé

Once downloaded, the JAR file must be added to the *plugins* folder inside the Protégé installation. The files in the installation directory look like following



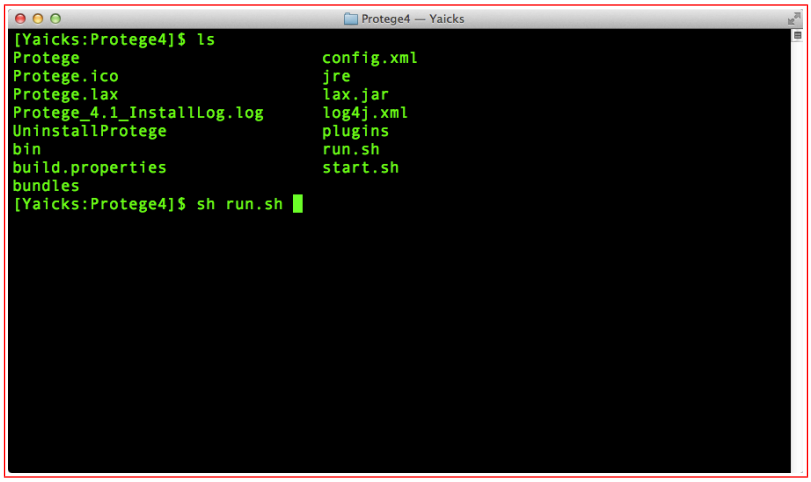
Inside the folder *plugins* you will find all the plug-ins that Protégé uses, and to enable a plug-in you just need to add the JAR file here and start the app. The folder for the plug-ins looks like following



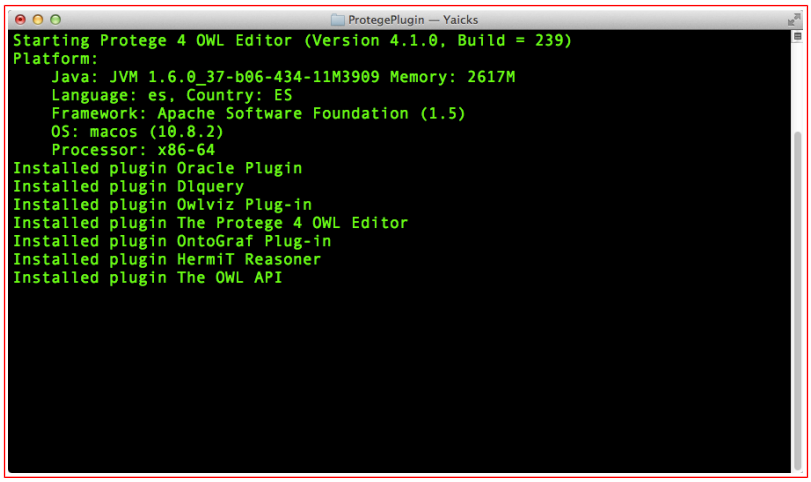
Executing Protégé

For Unix-based systems the running script is in the Protégé installation directory with the name *run.sh.* To run this script, type *sh run.sh* from a terminal as shown below.

Windows has an executable file which is added to the Start menu under All Programs.



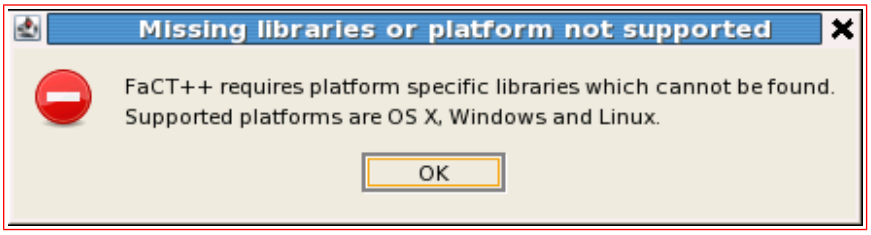
Once started, you will see information about your platform and a list showing which plug-ins are enabled. If the Oracle plug-in was successfully installed in Protégé, the list of plug-ins will display *Installed plugin Oracle plugin* as shown below



Once the welcome screen displays, the Oracle plug-in will add an *Open OWL Ontology from Oracle Database* and *Open OWL Ontology from Oracle NoSQL Database* options to this menu. At this point the plug-in is ready for use

The Fact++ Plug-in

Since the Fact++ plug-in does not run under Java, it only uses it to choose the correct C++ precompiled executable that this plug-in includes. There may be occasions in which your system will include an identifier that Fact++ will not be able to detect, causing Protégé to fail with the following error message.

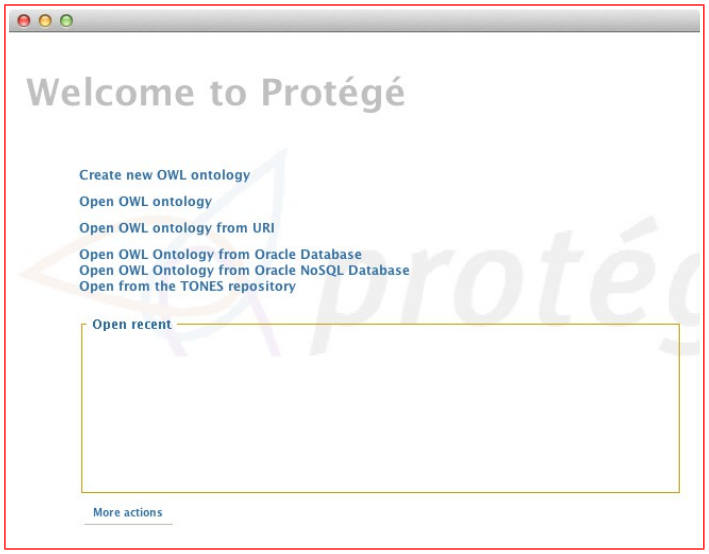


If this message appears, your system does not support the Fact++ plug-in and you will need to delete the *uk.ac.manchester.cs.owl.factplusplus.jar* file inside the *plugins* folder, which is shown below.

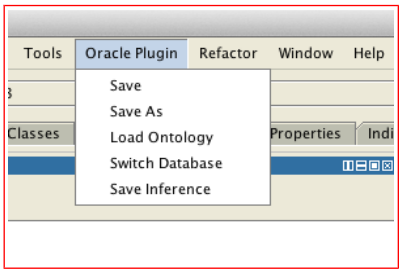
After deleting this JAR, you just need to run again Protégé in order for the plug-in to be disabled and to be able to use the application without errors.

Using the Oracle Plug-in

The Oracle plug-in adds two sets of options to Protégé, which are shown below.



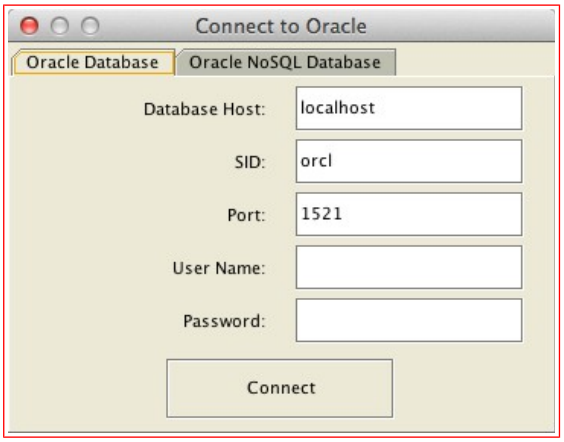
Welcome Screen Plug-in

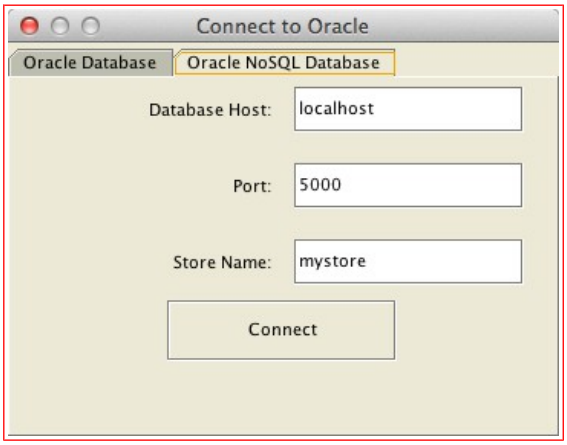


Menu Plug-in

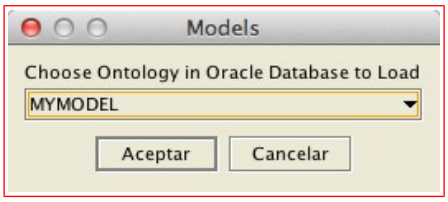
Welcome Screen

The welcome screen includes *Open OWL Ontology from Oracle Database* and *Open OWL Ontology from Oracle NoSQL Database,* which enables you to input the database connection properties.





After inputting the required information to connect to a valid Oracle database, you will be shown a dialog in which all available ontologies for the current users will be displayed, so that you can choose which ontology to load into Protégé.



After choosing the ontology, the load time will vary depending on your network connection speed and the size of the model.

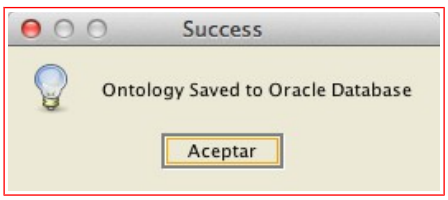
There is no data loading progress indicator in Protégé, which causes the welcome screen to seem frozen while the ontology is being loaded. After the ontology is loaded, the welcome screen will close and the main application will be displayed.

Oracle Plug-in Menu

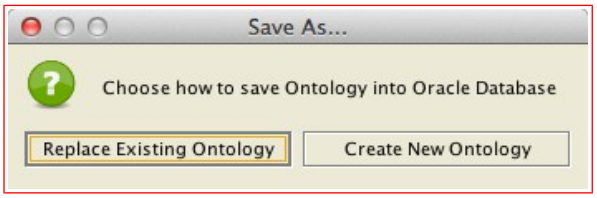
The main application of Protégé will have a menu with the options to *Save*, *Save As*, *Load Ontology, Switch Database* and *Save Inference*. When an ontology is loaded, the *Save* option will make an incremental edit saving in the database, whereas the *Save As* option will replace the ontology or create a new one.

After choosing *Save* or *Save As*, an information message will display explaining if the ontology was saved or if there was an error, as shown below.

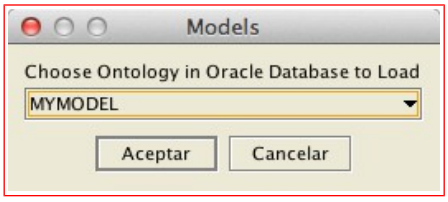
The message may be delayed depending on the size of your ontology, Oracle database / Oracle NoSQL Database configuration, and network speed.



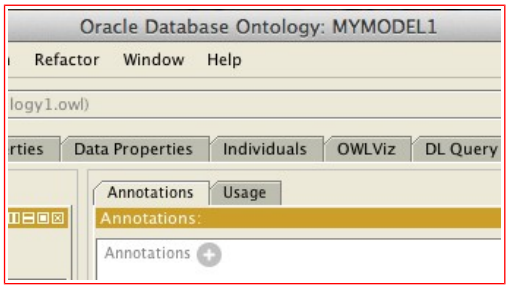
The *Save As* option displays a menu to choose between overwriting an existing ontology or creating a new one in the database.



When using the *Load Ontology* option, a dialog will display all the ontologies the current user can load from the database. After loading the ontology, the main application will display the name of the ontology in the window title bar.

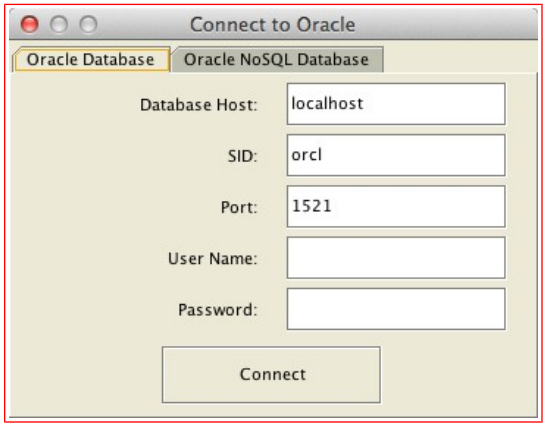


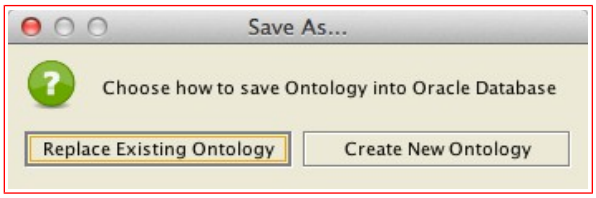
Plug-in Loading Dialog



Ontology Name on the Top Border

When Protégé has been started by choosing any Protégé option from the welcome screen, the first time you try to save the ontology you created, the database connection dialog is displayed so that you can connect to a database and save the ontology if you wish.
When this occurs, the Save and Save As options will replace or create a new ontology as shown below.





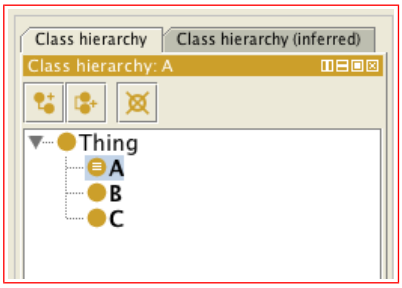
First Time Saving Dialogs – When Saving a Protégé Ontology to the Database

Using Protégé

Protégé is a visual editor for the Web Ontology Language. The main application is tab-based with options to edit *Classes*, *Object Properties*, *Data Properties*, *Individuals* and *Annotations*.
The Oracle plug-in provides compatibility with the database by providing conversion between Triples and OWL APIs in order for Protégé to be able to exchange information with it.

Editing Classes

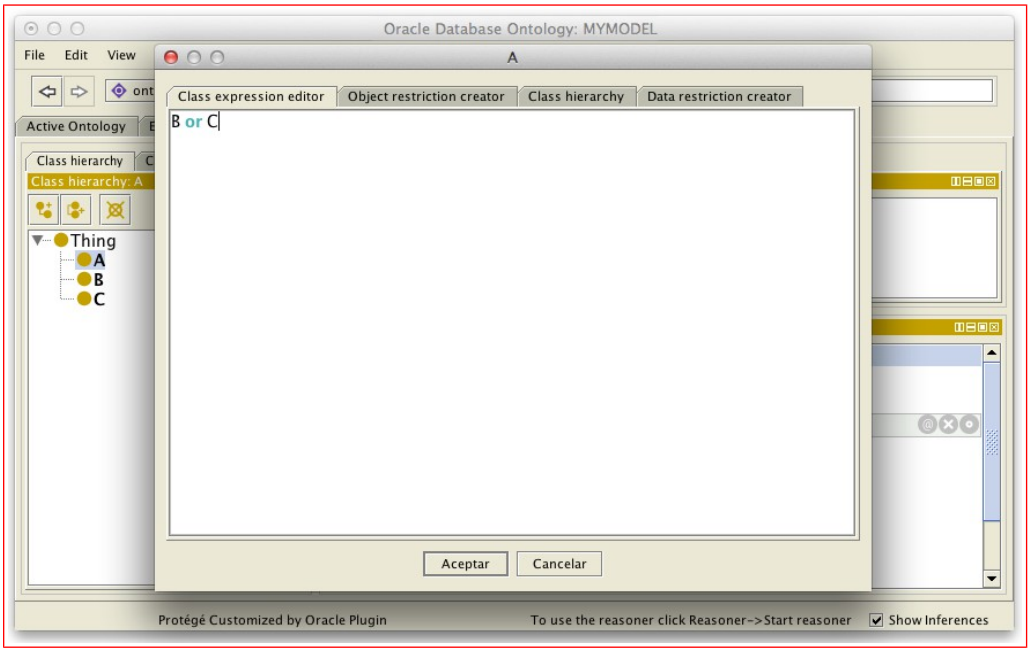
Once the main application is loaded, the classes can be edited in the *Classes* tab. All classes are a subclass of Thing (owl:Thing) and are selected for editing in the *Class hierarchy* as shown below.



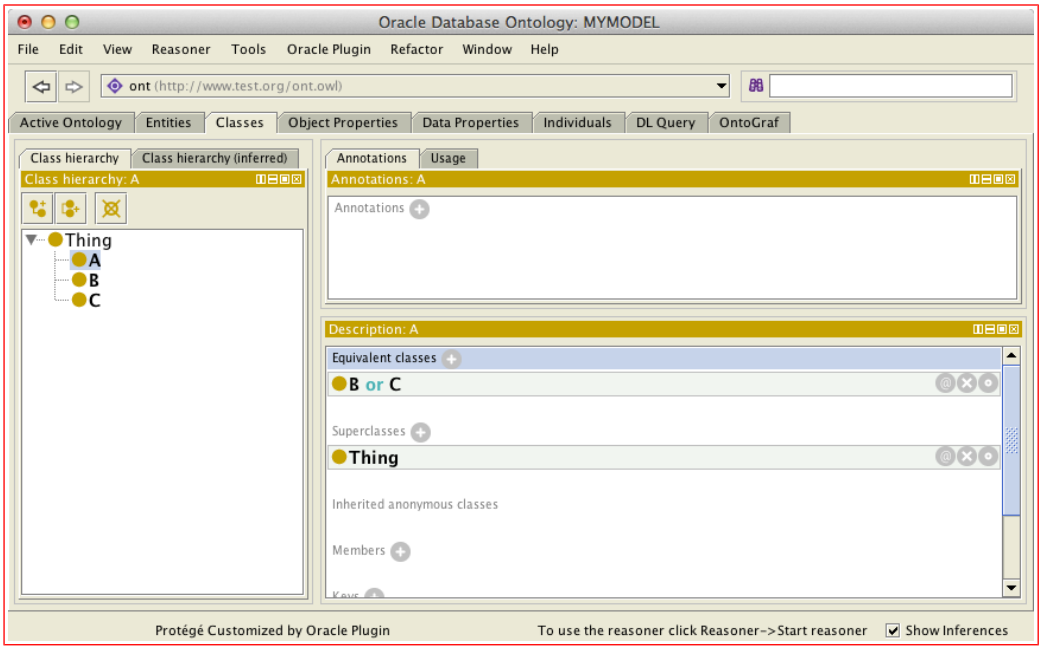
Here you have the options to add a subclass or sibling class or to remove one. Once a Class is added, you can select it to add annotations and descriptions of it.

Below is an example of how to add a union.

From the Description Panel, Press the Plus Sign on Equivalent Class of "A"



In the Class Expression Editor, Type "B or C"

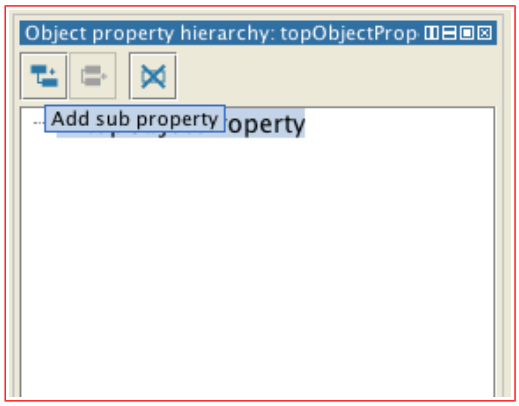


The Equivalence Will Be Displayed When Selecting The Class "A"

Editing Object Properties

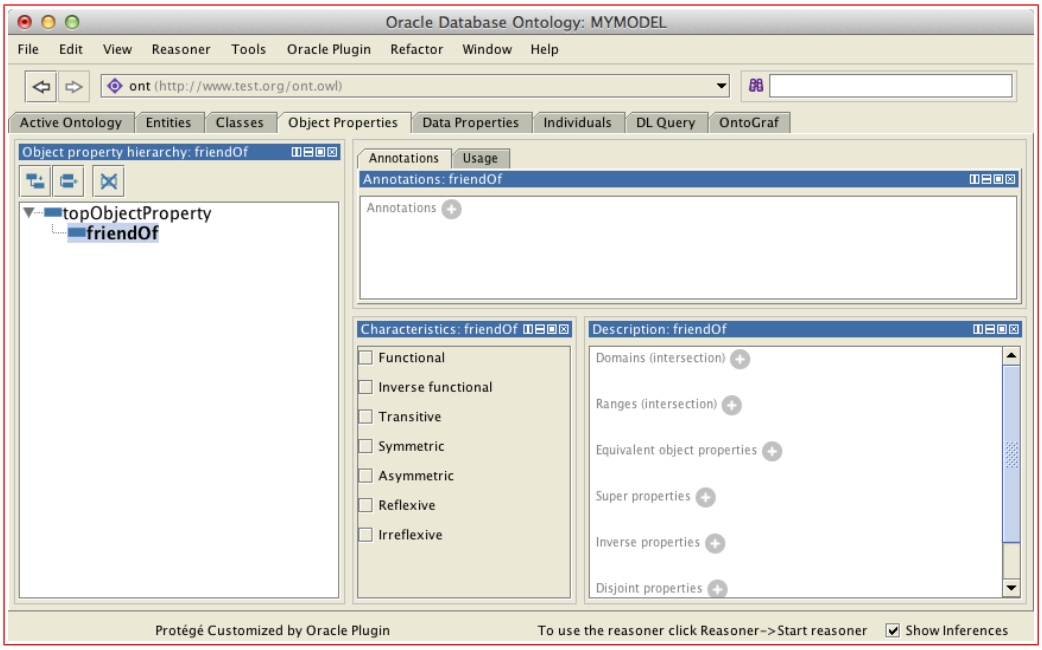
Object properties are edited in the *Object Properties* tab, similar to the classes tab you will have a hierarchy with topObjectProperty as the root of all object properties.

Below is an example of adding an object property.



From the Hierarchy Panel, Select the "topObjectProperty" and Then Press "Add sub property"

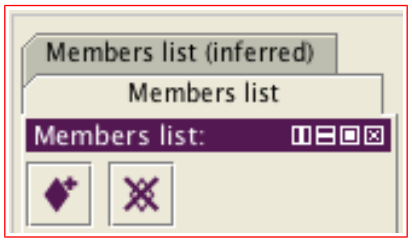
A Dialog Allows Us to Input the Desired Name for the Object Property



Once the Object Property is Added, We Can Edit the Description/Characteristics/Annotations

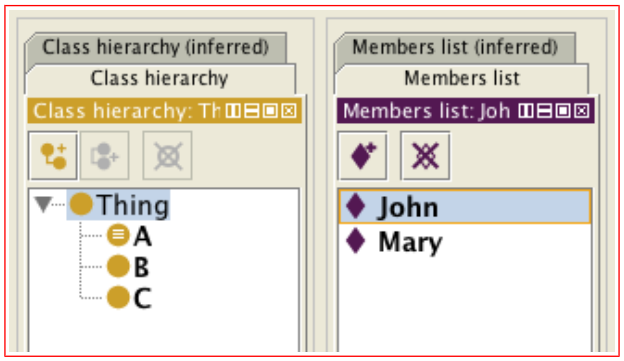
Editing Individuals

The Individuals tab differs from the rest of the tabs by adding a hierarchy for classes and a members list for the individuals. Individuals can belong to a specific class or be part of the Thing class. Once a class is selected, we can add/remove individuals from the members by using the Add/Delete buttons as shown below.

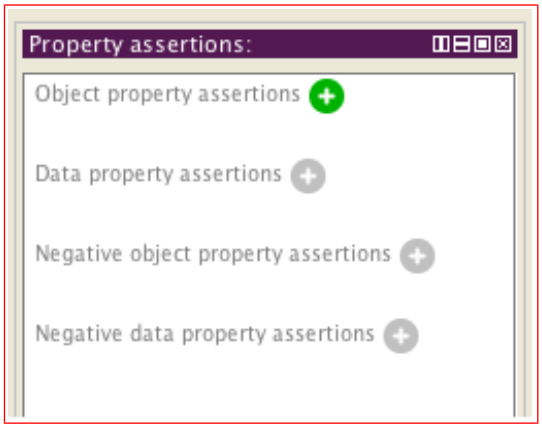


Once an individual is added, we can choose it from the members and edits its description, assertions and annotations.

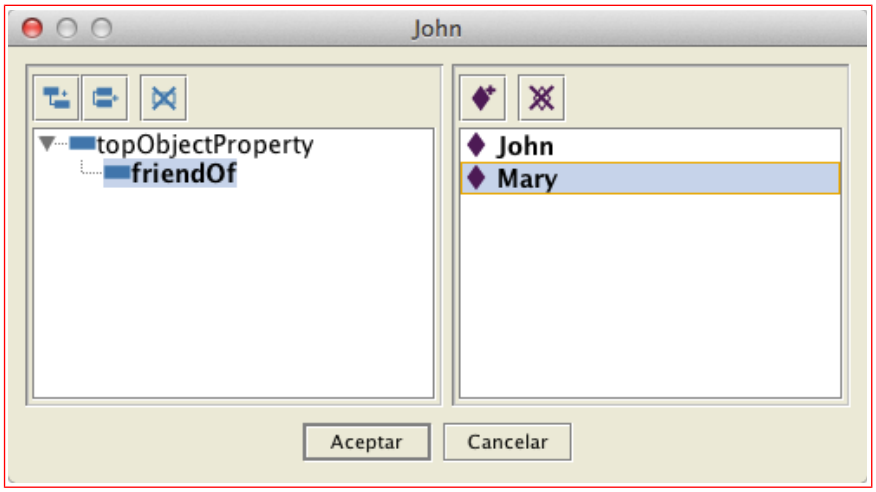
Below is an example of individuals with Object Property Assertions.



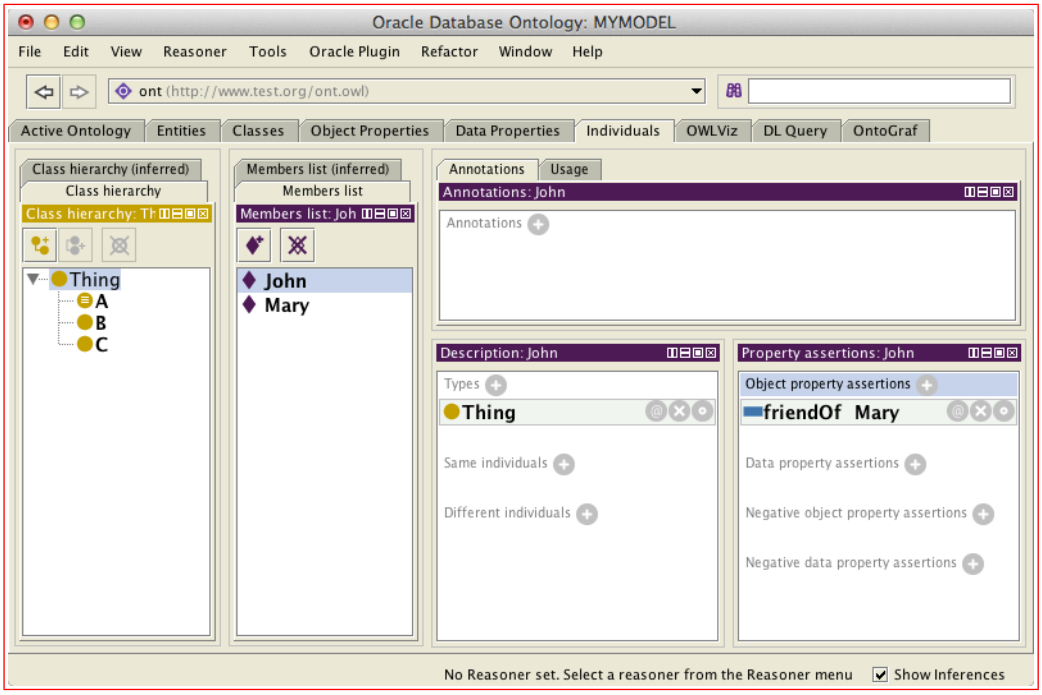
First Add Two Individuals to the Thing Class



On the Property Assertions of One of the Individuals, Press "Object property assertions"



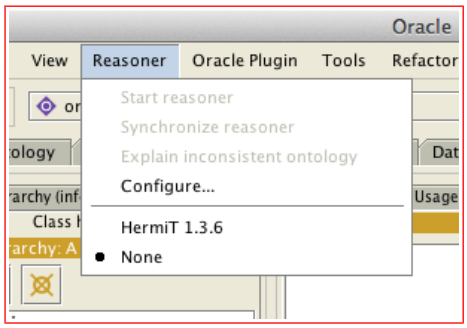
An Editor Lets You Choose an "Object Property" and "Individual" to Create the Assertion



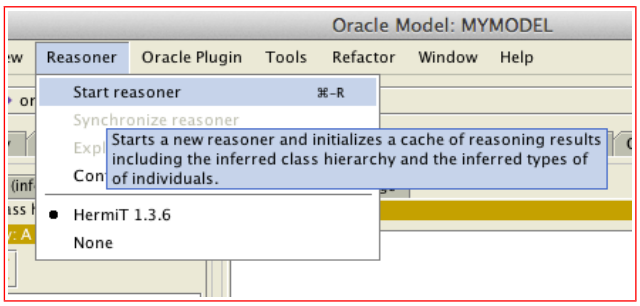
Once Selected, the Object Property Assertion Will be Displayed in the Assertion Panel

Using Protégé Reasoners

Protégé comes bundle with *HermiT* and *Fact++* for making inferencing on your ontology. In order to start you need to select the desired reasoner from the *Reasoner* menu.

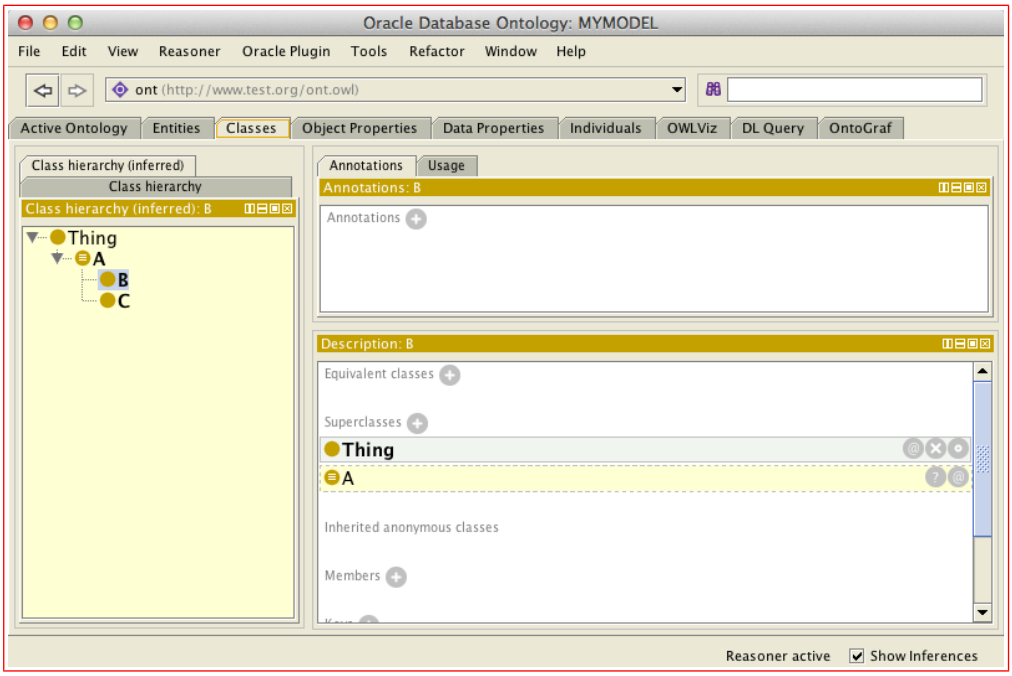


After selecting the reasoner, you can start the inferencing on the ontology by selecting the *Start reasoner* option from the *Reasoner* menu as shown below.

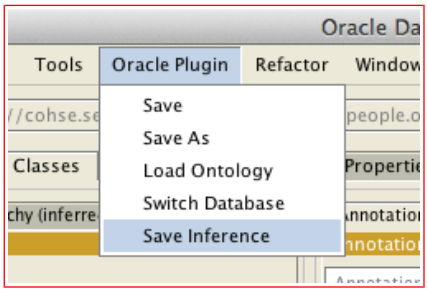


After the process is completed, the inferred classes and individuals will appear as a highlighted element in each of the description panels of their corresponding type. These elements cannot be edited, and you can easily access them by selecting the inferred class hierarchy and members list.

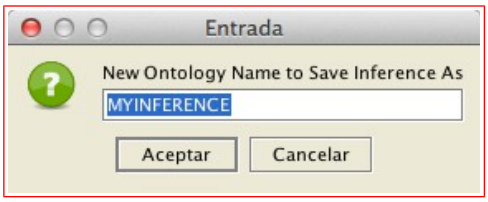
The Oracle plug-in will not save the inferred elements in the database.



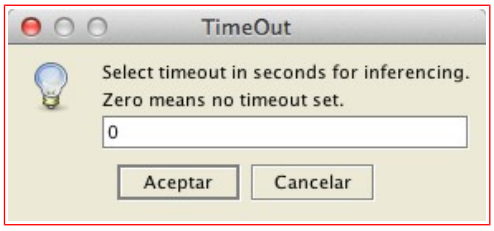
In order to save the inferred ontology you need to select the Save Inference from the plugin menu.
You need to start a reasoner first in order to save the inference.



You will be asked to input a name to save the inference as a new ontology into the database.



Before processing the inference, you will be asked to provide a timeout in seconds in order for the inference process to stop when this time is reach or set the timeout to zero to process the complete inference.



Once the inference process reaches its timeout or completes, a message dialog will be display giving feedback about the saving.