

NCI PROTÉGÉ

1.2

Workflow Manager's Guide



NATIONAL[®]
CANCER
INSTITUTE

Center for Bioinformatics

CONTENTS

About This Guide	1
Purpose	1
Audience	1
Topics Covered	2
Text Conventions Used	2
Credits and Resources	3
Application Support Contacts	4
Chapter 1	
Workflow Management Procedures	5
Overview of BiomedGT	5
About BiomedGT	5
<i>Purpose</i>	5
<i>Audience</i>	6
About the BiomedGT Wiki	6
Overview of Internal NCI Workflow	8
Primary Goals	8
Workflow Roles	8
Workflow Stages	9
About the NCI Workflow Tab	10
Workflow Procedures	12
Importing Packages from the BiomedGT Wiki	12
Assigning (or Reassigning) a Workflow Item	12
Chapter 2	
Baseline Comparison Procedures	15
Overview	15
About the Comparison Process	15
About the Prompt Plug-in	17
Preliminary Tasks	17
Notifying Protégé Users of Database Shutdown	17

Starting the Remote Protégé Client	17
Comparison Tasks	18
Enabling the Prompt Tab	18
Starting the Comparison	19
Interpreting the Comparison Results	21
<i>Table View Subtab</i>	21
<i>Tree View Subtab</i>	23
<i>Users Subtab</i>	25
Accepting and Rejecting Changes	27
<i>Accepting a Single Change</i>	27
<i>Accepting All Changes</i>	27
<i>Rejecting a Single Change</i>	27
<i>Rejecting All Changes</i>	28
Saving the Concept History	28
Disabling the Prompt Tab	28
Preparatory Tasks for the Next Comparison	29
Naming a Baseline Version	29
<i>Example: Naming the Exported File</i>	29
<i>Example: Specifying the owl:versionInfo Property Value</i>	29
Changing the Baseline Version Number for the Next Prompt Build	30
Exporting a Project File	30
Deleting Extraneous Information	31
<i>Change Ontology</i>	31
<i>Annotation Ontology</i>	31
<i>"Accepts" and Journal File</i>	31
<i>Deleting Both Ontologies</i>	32
Final Tasks	33

Chapter 3

Classification Procedures35

Overview	35
About the Classification Process	35
About the Pellet Reasoner	36
Preliminary Tasks	36
Notifying Protégé Users of Database Shutdown	36
Starting the Remote Protégé Client	37
Classification Tasks	38
Enabling the OWL Classes Tab	38
Starting the Reasoner	39
Asserting (Accepting or Rejecting) Current Classification Results	39

<i>Scenario 1: Accepting a Suggestion</i>	40
<i>Scenario 2: Rejecting a Suggestion</i>	41
Asserting (Accepting or Rejecting) Results from the Previous Classification	41
Saving the Current Classification Results as a Report	42
Closing Protégé Without Saving the Project	42
Final Tasks	43
Index	45

ABOUT THIS GUIDE

This section introduces you to the *NCI Protégé 1.2 Workflow Manager's Guide*.

Topics in this Section

- *Purpose* on this page
- *Audience* on this page
- *Topics Covered* on page 2
- *Text Conventions Used* on page 2
- *Credits and Resources* on page 3
- *Application Support Contacts* on page 4

Purpose

This guide is a reference for workflow managers using Protégé at the National Cancer Institute (NCI). It explains how to perform basic workflow management tasks, run a Prompt database comparison, and run the Pellet reasoner.

Audience

This guide is designed for NCI workflow managers who already have a solid understanding of the NCI Protégé semantic model and NCI Thesaurus editing processes and procedures.

Note: This guide is not intended for editors (modelers) who do not perform management tasks.

Topics Covered

This section provides an overview of the topics included in this guide.

- *Chapter 1* provides an overview of BiomedGT terminology and the workflow processes and procedures used by external collaborators and NCI terminology curators.
- *Chapter 2* explains how to use the Prompt comparison tool to compare two versions of the NCI Protégé database. It also includes a reference section that explains how to interpret the results of the comparison.
- *Chapter 3* explains how to classify the NCI Protégé database, accept or reject proposed changes, and prepare and save a classification report for distribution.

Text Conventions Used

This section explains conventions used in this guide. The various typefaces represent interface components, keyboard shortcuts, toolbar buttons, dialog box options, and text that you type.

Convention	Description	Example
Bold	Highlights names of interface elements such as option buttons, check boxes, drop-down menus, menu commands, command buttons, or icons.	Click Search .
URL	Indicates a Web address.	http://domain.com
Text in SMALL CAPS	Indicates a keyboard shortcut.	Press ENTER.
Text in SMALL CAPS + Text in SMALL CAPS	Indicates keys that are pressed simultaneously.	Press SHIFT + CTRL.
<i>Italics</i>	Highlights newly introduced terms; also highlights references to other documents, sections, figures, and tables.	This is known as <i>cloning</i> . See <i>Figure 4.5</i> .
<i>Italic monospaced type</i>	Represents text that you type.	In the New Subset text box, enter <i>Proprietary Proteins</i> .
Note:	Highlights information of particular importance	Note: This concept is used throughout the document.
{ }	Surrounds replaceable items.	Replace {last name, first name} with the Principal Investigator's name.

Credits and Resources

The following people contributed to the development of this guide.

NCI Protégé Development and Management Teams		
Development	Quality Assurance	Project and Product Management
Robert Dionne ²	Maria-Elena Hernandez ⁶	Frank Hartel ¹
Timothy Redmond ⁵	Steven Hunter ⁷	Gilberto Fragoso ¹
Tania Tudorache ⁵	Tracy Safran ³	Sherri De Coronado ¹
Natasha Noy ⁵		Laura Roth ²
Kim Ong ⁴		Carol Creech ²
Iris Guo ⁴		
Chris Callendar ⁶		
Documentation	Systems and Application Support	User Community Resources
Edward VanArsdall ²	Tracy Safran ³	John Bradsher ²
	Robert Wynne ²	Liz Hahn-Dantona ²
		Nicole Thomas ²
¹ National Cancer Institute Center for Bioinformatics (NCICB)	² Lockheed Martin	³ Science Application International Corporation (SAIC)
⁴ Northrup Grumman	⁵ Stanford Medical Informatics (SMI)	⁶ University of Victoria
⁷ Ekagra Software Technologies		

Application Support Contacts

To obtain general information about NCI Protégé, receive support, or report a bug, contact NCICB Application Support.

<i>Support Option</i>	<i>Additional Information</i>
For NCI Protégé Users	
GForge	<p>If you are an NCI Protégé user, post all bug reports and requests for enhancements on GForge using the following URL: https://gforge.nci.nih.gov/tracker/?group_id=174</p> <p>For instructions on reporting bugs, see the <i>NCI Protégé 1.2 Editor's Guide</i>.</p>
Administrative tasks for NCI Protégé	<p>EVS Operations Support can provide the following services:</p> <ul style="list-style-type: none"> • Loading a database • Creating a user account • Creating or configuring a Protégé project • Installing or reinstalling the Protégé client on a user's machine <p>Note: If you need any of these services, send your request to the e-mail address under the For General Users section (below).</p>
For General Users	
E-mail	<p>Write to ncicb@pop.nci.nih.gov and include the following information:</p> <ul style="list-style-type: none"> • Your contact information, including your phone number • The name of the application that you are using • The URL (for Web-based applications) • A description of the problem and the steps required to recreate it • The text of any error messages you have received.
Web	<p>NCICB support: http://ncicbsupport.nci.nih.gov/sw/</p> <p>NCI Protégé online support forums: https://gforge.nci.nih.gov/forum/?group_id=174</p>
Telephone	<p>Local: 301-451-4384 Toll-free: 888-478-4423</p> <p>Telephone support is available Monday through Friday, 8 a.m. - 8 p.m. Eastern time, excluding government holidays.</p>

CHAPTER 1

WORKFLOW MANAGEMENT PROCEDURES

This chapter provides an overview of BiomedGT and the workflow processes and procedures used by external collaborators and NCI terminology curators.

Topics in this Chapter

- [Overview of BiomedGT](#) on this page
- [Overview of Internal NCI Workflow](#) on page 8
- [Workflow Procedures](#) on page 12

Overview of BiomedGT

About BiomedGT

Purpose

Biomedical terminologies are typically large, standalone terminologies that cover many domains in one structure and use a centralized editing model. Recognizing that this type of terminology no longer adequately supports the needs of the biomedical research community, the NCI Center for Bioinformatics (NCICB) and the cancer Biomedical Informatics Grid (caBIG™) have launched the Biomedical Grid Terminology (BiomedGT).

BiomedGT is an open, collaboratively developed terminology for translational research. This new terminology builds on the strengths of the NCI Thesaurus, including concept orientation, description logic, and public accessibility. While the initial content of BiomedGT is based on the NCI Thesaurus, the terminology has been restructured to facilitate open content development. The goal is to evolve BiomedGT into a set of smaller federated sub-terminologies called *domain namespaces*, which cover specific areas of biomedical knowledge. As experts in the relevant research communities stabilize the content of the various domain namespaces, new content will be incorporated into the description logic-based, federated BiomedGT by the NCICB EVS and volunteer editors.

BiomedGT is intended to be the first of a new generation of biomedical terminologies that are created using an open content development approach. Such an approach empowers the wider biomedical research community to directly participate in extending and refining the terminology upon which they depend. The approach also enables maximum community participation without requiring that contributors learn complex terminology curation software and without sacrificing the quality assurance standards needed to manage biomedical terminologies.

Audience

BiomedGT is intended to address the needs of biomedical researchers who work in both basic and clinical research. Researchers can acquire content by

- downloading it in multiple formats such as OWL and RRF;
- using Web-enabled applications; or
- using Application Programming Interfaces (APIs).

About the BiomedGT Wiki

The BiomedGT wiki (<http://biomedgt.org>) is a Web-based, collaborative terminology authoring tool being developed by the NCICB and Apelon, Inc. *Figure 1.1* shows the home page of the wiki.

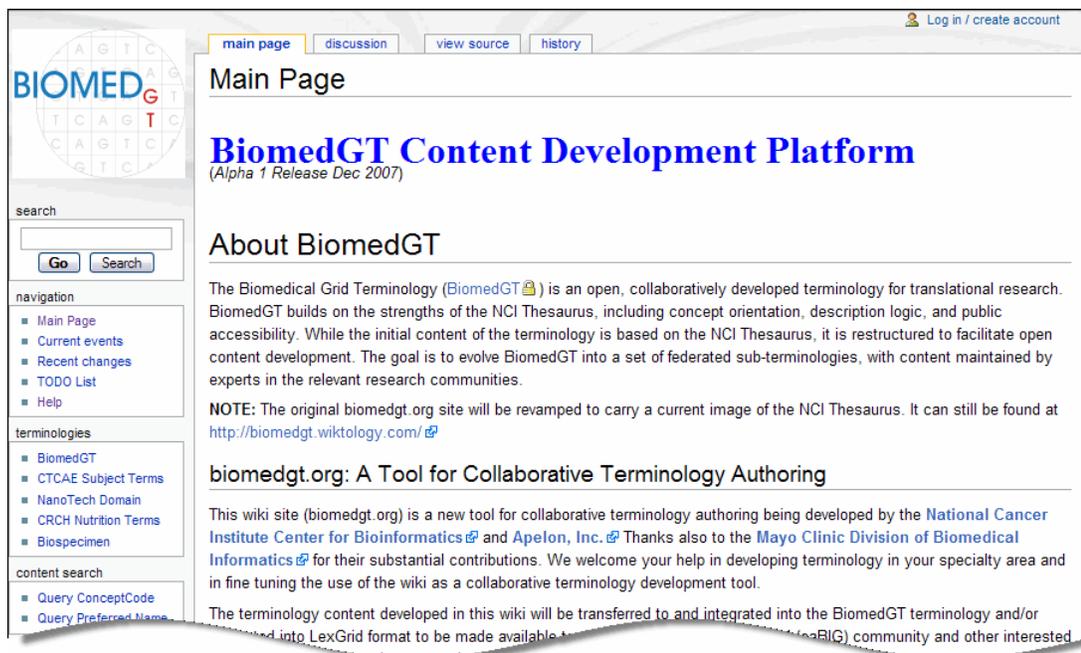


Figure 1.1 BiomedGT Home page

The BiomedGT wiki pulls terminology data from a read-only database and displays it on a Web page. This facilitates the process described in [Table 1.1](#).

Stage	Action	Responsibility
1.	Review the read-only, published BiomedGT terminology.	Outside collaborator
2.	Using the provided Web form, submit a <i>structured</i> or <i>unstructured</i> proposal for any of the following reasons: <ul style="list-style-type: none"> • To suggest changes to existing concepts (structured) • To suggest the addition of new concepts (structured) • To direct NCI curators to a specific wiki discussion page by providing the URL (unstructured). Structured proposals are also called <i>packages</i> because they often include batches of concepts. Unstructured proposals typically refer to a wiki discussion page that might discuss specifics about a concept and then make a recommendation.	Outside collaborator
3.	Import packages and incorporate them into the vocabulary as needed. This stage has its own internal workflow processes and procedures as described in Overview of Internal NCI Workflow on page 8.	NCI BiomedGT curators
4.	Communicate with submitting collaborators to clarify suggestions or explain the outcome of decisions regarding proposals.	NCI BiomedGT curators
5.	Publish updated read-only database and display it on the Web page.	BiomedGT administrator

Table 1.1 Workflow process: BiomedGT to Protégé

For more information about BiomedGT, see the following online resources:

- BiomedGT project on GForge:
<http://gforge.nci.nih.gov/projects/biomedgt/>
- EVS section of the NCICB web site:
http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/vocabulary
- BiomedGT wiki site:
<http://biomedgt.org>

Overview of Internal NCI Workflow

As biomedical research community experts increasingly use the BiomedGT wiki for collaborative terminology development, NCI curators need a managed, organized workflow process. This section discusses the process established for the NCI.

Primary Goals

Following are the primary goals of the BiomedGT/Protégé workflow:

- To provide capability that supports workflow manager and modeler roles, including assignments, tracking, auditing, and reporting
- To provide a means for extracting new terms from the BiomedGT wiki and incorporating them into assignments for NCI Protégé modelers
- To provide status and auditing reports to help management assess modeling progress.

Workflow Roles

The internal workflow for NCI Protégé consists of two primary roles:

- **Manager**

A manager oversees the coordination and distribution of work assignments. The assignments may originate from the BiomedGT wiki or from other members of the editing team.

The manager performs the following tasks:

- Imports packages from the BiomedGT wiki
- Creates assignments for editors
- Tracks assignments
- Incorporates completed assignments into the NCI Thesaurus.

Note: The manager may also perform routine editing tasks.

- **Modeler (Editor)**

A modeler is an NCI editor who performs routine editing tasks. Modelers cannot import packages, create assignments, or delete assignments. However, they can make *suggestions*, which are similar to assignments but which do not become assignments until a manager has reviewed them.

The modeler performs the following tasks:

- Receives assignments and accepts or rejects them
- Incorporates new or changed terminology into the NCI Thesaurus using the NCI Protégé work environment
- Maintains responsibility for the assigned concept, including any follow-up communication to the collaborator who submitted the concept for consideration
- Marks assignments as complete and submits them to the manager.

Workflow Stages

Table 1.2 provides an overview of the internal workflow stages and the responsible role for each stage.

Note: Each submitted item is called a *workflow item* until it has been assigned to a modeler. It then becomes an *assignment*.

Stage	Action	Responsibility
1.	Log into Protégé.	Manager
2.	Using the NCI Workflow tab, import packages from the BiomedGT wiki into Protégé.	Manager
3.	Create workflow items with a status of CREATED.	System
4.	Review each new workflow item and assigns it to a modeler.	Manager
5.	Change the status of the workflow item to ASSIGNED.	System
6.	Log into Protégé.	Modeler
7.	Check the NCI Workflow tab for assignments.	Modeler
8.	Accept an assignment.	Modeler
9.	Change the status of the assignment to ACCEPTED.	System
10.	Examine the proposed and existing (if any) concepts.	Modeler
11.	Perform one of the following tasks: <ul style="list-style-type: none"> Edit and save the existing concept (if any); if completed, mark the assignment as COMPLETED. Add notes to the assignment regarding the proposed concept and existing concept (if any); if completed, mark the assignment as COMPLETED. Reject the assignment. 	Modeler
12.	Change the status of the assignment to UPDATED (if not marked COMPLETED), COMPLETED, or REJECTED, depending on the action in step 11.	System
13.	Review completed and rejected assignments; address rejected assignments according to the nature of the proposal.	Manager

Table 1.2 BiomedGT wiki process for curating terminology

Note: In addition to work items generated by BiomedGT wiki imports, the manager also handles work items that have been submitted internally by modelers. The system assigns a status of SUGGESTED to internally submitted work items.

About the NCI Workflow Tab

The **NCI Workflow** tab is the major workflow component used by both managers and modelers. The tab layout is a table grid format, with each row representing a workflow item or assignment (*Figure 1.2*).

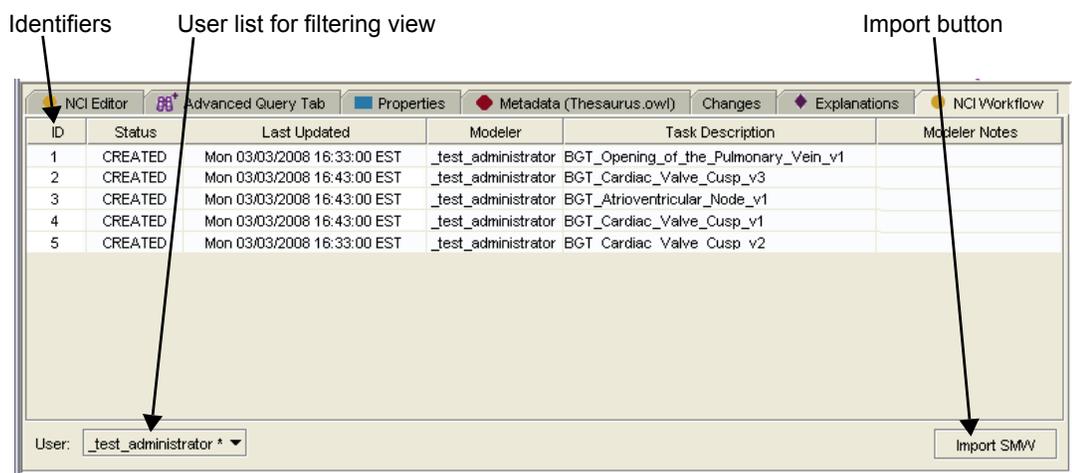


Figure 1.2 NCI Workflow tab

Protégé determines what the **NCI Workflow** tab displays based on the privileges of the user who is currently logged in:

- Managers can see all assigned and unassigned workflow items and can identify the assignee and status of each assigned item. They can also filter the assignments by user.
- Modelers see only workflow items that have been assigned to them.

Table 1.3 describes the fields that identify each workflow item and assignment.

Field	Description
ID	Shows a sequential number that is assigned to each workflow item. Once a number is assigned to an item, it remains the same, even if the same item is re-imported.
Status	Shows the current status of a workflow item or assignment: <ul style="list-style-type: none"> • CREATED: The manager has imported a workflow item but has not yet assigned it. • SUGGESTED: A modeler has created a workflow item that is relevant to an existing concept. • ASSIGNED: The manager has assigned a workflow item to a modeler but the modeler has not yet accepted it. • ACCEPTED: The modeler has accepted the assignment. • UPDATED: The modeler saved a change to a class or added a note but did not mark the assignment as COMPLETED. • COMPLETED: The modeler has completed the assignment.

Table 1.3 NCI Workflow tab fields

Field	Description
Last Updated	Shows the weekday, date, and time in the following format: Fri 03/07/2008 16:34:00 EST
Modeler	Initially shows the name of the user (manager) who imported the workflow item. Once assigned, this field displays the name of the assignee.
Task Description	For structured proposals, shows the name of a concept or a URL pointing to a wiki discussion page. Concept names are preceded by the letters BGT to note that the concept has been imported from the BiomedGT wiki.
Modeler Notes	Shows the most recent note added to a workflow item or assignment, followed by an ellipsis (...). If you hover the mouse pointer over the date and time, pop-up text shows the note text starting with the most recent entry.

Table 1.3 NCI Workflow tab fields (Continued)

The grid area of the **NCI Workflow** tab supports right-click commands. [Table 1.4](#) describes the available commands. Some commands may appear as unavailable depending on your user privileges.

Command	Description
Edit Concept	Displays the proposed concept and existing concept (if any) on the Workflow subtab. You can then edit, add notes to, or reject the concept.
Accept Assignment	Enables you to accept an assignment.
Modify Assignment	Opens a window in which you can add notes and view previous notes.
Complete Assignment	Enables you to change the status of an assignment to COMPLETED.
Delete Assignment	Enables you to delete an assignment, depending on your user privileges and the status of the assignment: <ul style="list-style-type: none"> Managers can only delete assignments with a status of CREATED, SUGGESTED, or COMPLETED. Modelers can only delete SUGGESTED assignments.
Cancel	Closes the shortcut menu.

Table 1.4 NCI Workflow tab - right-click commands

Workflow Procedures

This section provides the steps for the main procedures that NCI workflow managers typically perform. Most of the procedures originate from the NCI Workflow tab.

Note: For more information about workflow manager procedures in relation to the overall NCI internal workflow process, see *Overview of Internal NCI Workflow* on page 8.

Topics in this Section

- [Importing Packages from the BiomedGT Wiki](#) on this page
- [Assigning \(or Reassigning\) a Workflow Item](#) on page 12

Importing Packages from the BiomedGT Wiki

To assign new workflow items from outside collaborators, you first need to import a package from the BiomedGT wiki. The NCI Workflow tab provides a button for this purpose.

To import a package from the BiomedGT wiki, follow these steps:

1. Log in to Protégé.
2. Click the **NCI Workflow** tab.
3. Click the **Import SMW** button in the lower right corner of the tab.

One of two messages appear, depending on whether packages are available on the wiki:

- If packages are available, a confirmation message appears. The message lists the ID numbers of any new assignments.
- If no packages are available, a message indicates that there is nothing available to import.

Note: For each newly imported item, the **Modeler** column shows the name of the user who imported that item.

Assigning (or Reassigning) a Workflow Item

After importing items from the BiomedGT wiki, you need to assign those items to modelers.

To assign a workflow item to a modeler, follow these steps:

1. Click the **NCI Workflow** tab.
2. Locate the item that you want to assign.

Tip: If you are assigning an item that you imported, the items shows your name in the **Modeler** column. To see only items with your name, select your name from the **User** drop-down list in the lower left corner of the **NCI Workflow** tab.

3. In the **Modeler** column, select a different name from the drop-down list.

Note: These steps also apply to reassigning a workflow item to a different user.

From Here...

- *Baseline Comparison Procedures* on page 15
- *Classification Procedures* on page 35

CHAPTER 2

BASELINE COMPARISON PROCEDURES

This chapter explains how to use the Prompt comparison tool to compare two versions of the NCI Protégé database. It also includes a reference section that explains how to interpret the results of the comparison.

Topics in this chapter:

- [Overview](#) on this page
- [Preliminary Tasks](#) on page 17
- [Comparison Tasks](#) on page 18
- [Preparatory Tasks for the Next Comparison](#) on page 29
- [Final Tasks](#) on page 33

Overview

As a workflow manager, you will need to regularly compare two versions of the Protégé database, examine the results, and resolve any conflicts. This section introduces you to the process and the tool that you will use.

About the Comparison Process

An ontology maintained in Protégé is known as a *project* and is stored as a file with a .pprj extension. During each editing cycle, editors use the master Protégé project, which is maintained by Operations staff. This master project represents the current state of the database.

Each time you run a baseline comparison, you export a copy of the master project file and use it as a baseline during the next comparison cycle. This enables you to identify changes to the database and accept or reject them.

As editors make changes to the database, their changes are stored in a file called an *annotation ontology*. This file serves as an audit trail of changes as they take place. During a comparison, Prompt uses this file to determine changes between the current

project file and the most recent baseline file. Once you accept or reject the changes in the annotation ontology, you close Protégé and delete the annotation ontology and its related files to remove the record of changes that you have already reviewed. You then re-open Protégé and save the current project. This ensures that the current project builds a new change ontology, which serves as a “clean slate” for the next editing cycle.

Table 2.1 summarizes the process stages for comparing two versions of the Protégé database.

Stage	Action	Responsibility
<i>Preliminary Tasks</i> on page 17		
1.	Send e-mail or chat to alert modelers that the NCI Protégé database will be shut down and that they need to log out.	Manager
2.	Log out of the database.	Modeler
3.	Log into the Protégé host computer.	Manager
4.	Shut down the Protégé server.	Manager
5.	(Optional) Shut down the Explanation server	Manager
6.	Start the remote Protégé client and connect to the BiomedGT project.	Manager
<i>Comparison Tasks</i> on page 18		
7.	Configure the remote Protégé client to display the Prompt tab.	Manager
8.	Using Prompt, compare the current version of the database project (.pprj file) against the most recent baseline project.	Manager
9.	Accept or reject changes.	Manager
10.	Save the concept history.	Manager
11.	Disable the Prompt tab.	Manager
<i>Preparatory Tasks for the Next Comparison</i> on page 29		
12.	Change the baseline version shown on the Metadata tab.	Manager
13.	Export a copy of the current project file for use as a baseline for the next database comparison.	Manager
14.	Exit the remote Protégé client without saving the project.	Manager
15.	Delete all extraneous information, including the change ontology, annotation ontology and related files, and other specific files.	Manager
16.	Restart the Protégé client, save the project to create a new change ontology, then exit the client.	Manager
<i>Final Tasks</i> on page 33		
17.	Restart the Explanation server (if you previously shut it down).	Manager
18.	Start the Protégé server and verify that it is running.	Manager
19.	Verify that the Explanation tab is communicating with the Explanation server.	Manager
20.	Log off of the Protégé host computer.	Manager

Table 2.1 Database comparison process

About the Prompt Plug-in

In NCI Protégé, the **Prompt** tab operates the Prompt plug-in, which enables you to compare two different versions of the same ontology, examine the results, and accept or reject changes. When you run a comparison, the current ontology is considered the new version, and the external file that you specify is treated as the older version. This helps you to establish a version control system.

Prompt also has other capabilities that are currently not in use at the NCI. For more information, visit <http://protege.cim3.net/cgi-bin/wiki.pl?Prompt>.

Preliminary Tasks

The following procedures correspond to steps 1 through 6 in *Table 2.1* (page 16).

- [Notifying Protégé Users of Database Shutdown](#) on this page
- [Starting the Remote Protégé Client](#) on this page

Notifying Protégé Users of Database Shutdown

Prior to performing a baseline comparison, follow these steps:

1. Using e-mail or chat, announce that the database will be shut down.
2. Log out of the Protégé client on your own machine.

Starting the Remote Protégé Client

Note: In the following instructions, the current NCI Protégé directory is represented as `Protege.Server-x.y.z`. When typing commands, substitute the directory name with the current Protégé version number (**Example:** `Protégé.Server-1.2.1.21`).

To start the Protégé client on the remote server, follow these steps:

1. Log in to the Protégé host computer:
 - a. Open Reflection X.
 - b. Enter your user name and password.
2. Shut down the Protégé server:
 - a. Type the following command at the command prompt:


```
cd /usr/local/protege/Protege.Server-x.y.z
```
 - b. Press ENTER.
 - c. Type the following command:


```
./shutdown_protege_server.sh localhost:2633
```
 - d. Press ENTER.
3. (Optional) Shut down the Explanation server:
 - a. Type the following command at the command prompt:


```
cd /usr/local/protege/Explanation.Server-x.y.z
```

- b. Press ENTER.
- c. Type the following command:

```
./stop_explanation_server.sh
```
- d. Press ENTER.
4. Start the remote Protégé client:
 - a. Type the following command at the command prompt:

```
cd /usr/local/protége/Protege.Client-x.y.z
```
 - b. Press ENTER.
 - c. Type the following command:

```
./run_protége.sh
```
 - d. Press ENTER.
 - e. When prompted to select a project, select **BiomedGT**.

Comparison Tasks

The following procedures correspond to steps 7 through 11 in *Table 2.1* (page 16).

- [Enabling the Prompt Tab](#) on this page
- [Starting the Comparison](#) on page 19
- [Interpreting the Comparison Results](#) on page 21
- [Accepting and Rejecting Changes](#) on page 27
- [Saving the Concept History](#) on page 28
- [Disabling the Prompt Tab](#) on page 28

Enabling the Prompt Tab

The **Prompt** tab enables you to compare the current Protégé project file to a specified baseline file (usually the most recent version of the database). This tab is not enabled by default.

To enable the **Prompt** tab, follow these steps:

1. Select the following menu command: **Project > Configure...**
The **Configure File** window opens.
2. In the **Visible** column, check the box to the left of the **Prompt** widget.
3. Click **OK** to close the **Configure File** window.

The **Prompt** tab now appears in the main Protégé window. The tab is positioned on the far right (*Figure 2.1*).

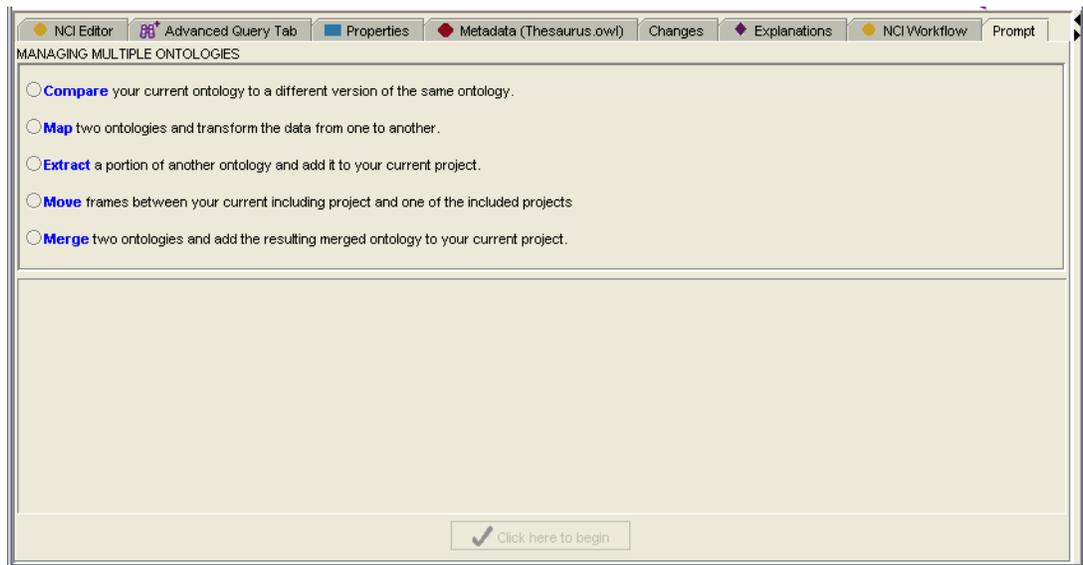


Figure 2.1 Prompt tab

Starting the Comparison

Prerequisite task: *Enabling the Prompt Tab* on page 18

To initiate a baseline comparison using the **Prompt** tab, follow these steps:

1. Click the **Prompt** tab.
2. Select the following option (if it is not already selected):
Compare your current ontology to a different version of the same ontology.
 New text entry fields and check boxes appear in the bottom half of the tab.
3. In the field labeled **Choose the version to compare with the current project**, do one of the following:
 - Enter the path and name of the file that represents the most recent baseline (*Figure 2.2*); or
 - Select a file by clicking the **Browse for File** button  on the right, just above the field (*Figure 2.2*).
4. Leave the remaining settings as they are:
 - **Display changes for included frames** is checked by default.
 - The field labeled **Select a slot containing a concept ID** is empty.

Note: Although you can theoretically use this field and its accompanying check box to add a slot and specify that a comparison be run only on a specific concept, the feature does not currently work as expected.

- At the bottom of the tab, click the button labeled **Click here to begin** (Figure 2.2).

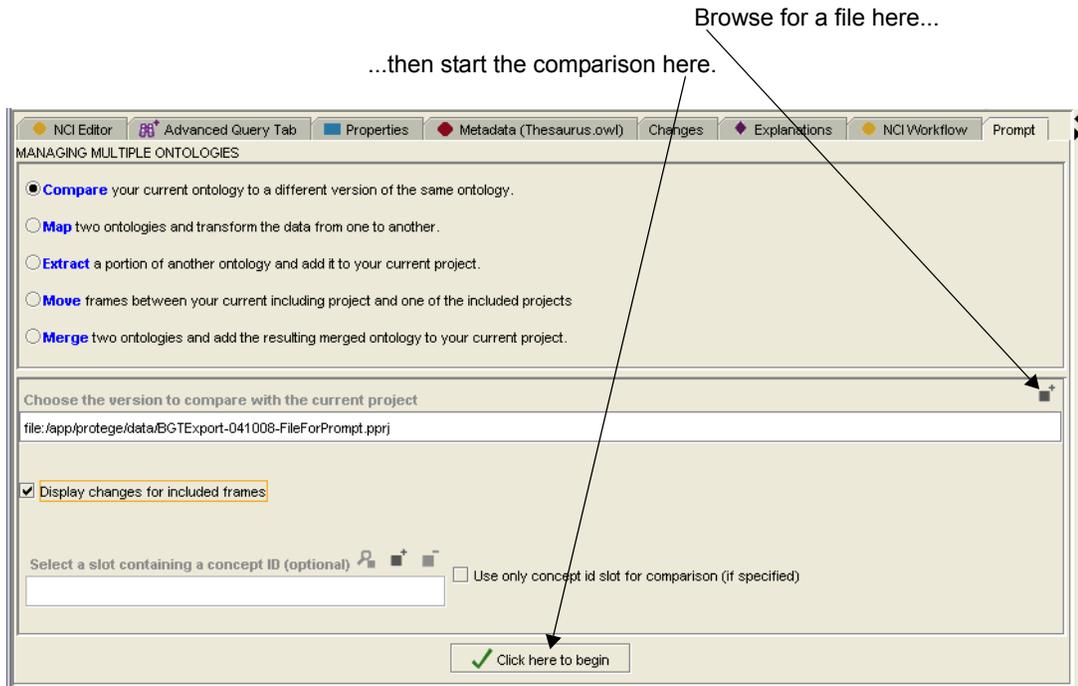


Figure 2.2 Comparing two versions of the database

A small status window appears in the foreground. When the comparison finishes running, the window disappears and a set of subtabs appears. The prominent **Tree View** subtab shows the two ontologies in a single tree hierarchy (Figure 2.3).

For more information about reading the subtab display, see *Interpreting the Comparison Results* on page 21.

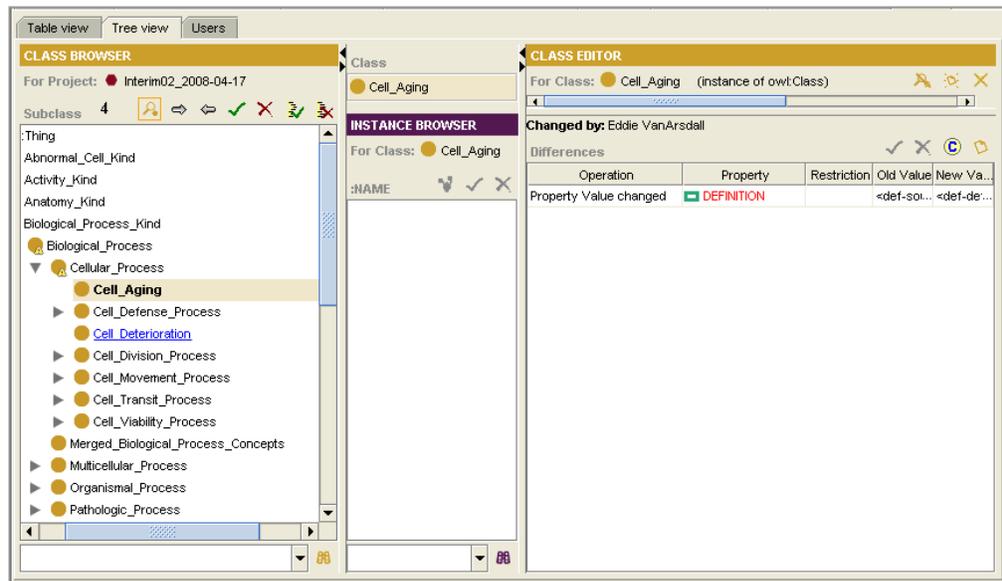


Figure 2.3 Prompt comparison results - Tree View subtab

Interpreting the Comparison Results

This section explains the three **Prompt** subtabs used for interpreting database comparison results. The subtabs are discussed in the order that they appear on the interface:

- [Table View Subtab](#) on this page
- [Tree View Subtab](#) on page 23
- [Users Subtab](#) on page 25

Table View Subtab

The **Table View** subtab ([Figure 2.4](#)) shows all of the frames in the two compared ontologies and lists their similarities and differences.

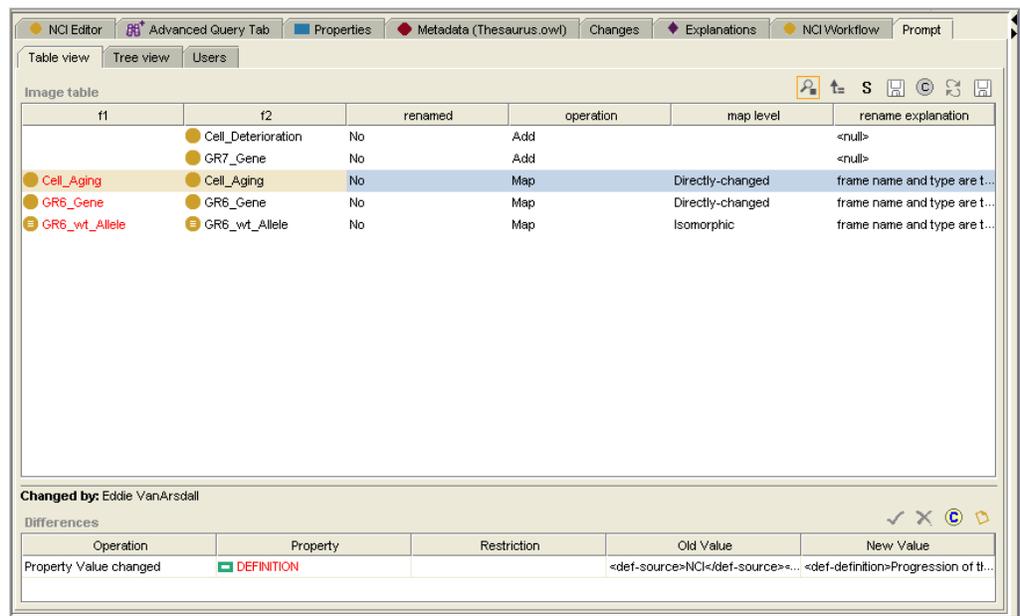


Figure 2.4 Table View subtab

[Table 2.2](#) describes the upper and lower panels of the **Table View** subtab.

Panel/Column	Description
Image Table	
f1	Represents the frames in the original ontology.
f2	Represents the frames in the changed ontology.
renamed	Indicates whether a class was renamed (Yes or No).
operation	Indicates the nature of the change: <ul style="list-style-type: none"> • Add: A new class was added to the ontology. • Map: The relationship between the class and certain other classes was changed.

Table 2.2 Table View subtab - panel descriptions

Panel/Column	Description
map level	Indicates the type of change: <ul style="list-style-type: none"> • Directly-changed: A direct change to the class that does not affect any other class. • Isomorphic change: A change to the child class that causes a change in its parent class.
rename explanation	Provides an explanation for any classes that have been renamed; otherwise displays the following message: <i>frame name and type are the same.</i>
Differences	
Operation	Explains the nature of the change (for example, <i>Property Value changed</i>).
Property	Describes which property was changed (if any).
Restriction	Describes which restriction was changed (if any).
Old Value	Shows the value before the change.
New Value	Shows the new value resulting from the change.

Table 2.2 Table View subtab - panel descriptions (Continued)

Table 2.3 describes the buttons used in each of the **Table View** subtab panels.

Button	Description
Image table	
View Frames 	Opens a window in which you can view annotations, asserted conditions, and disjoints. The window is essentially a standalone version of the Class Editor.
Back-references for Frames 	Opens a window that shows references (frames, slots, and facets) to the selected class and enables you to drill down to further references.
Show Sources 	Opens a window in which you can view Source classes, Source slots, and Source instances.
Save to File 	Saves the change as a tab-delimited .diff file. Note: This is <i>not</i> the button used to save the NCI concept history.
Change Statistics 	Opens a window showing a summary of change statistics, including additions, deletions, splits, merges, direct changes, and total changes.
Refresh Table 	Refreshes the Image table panel.
Save in NCI DB Format 	Saves the concept history and stores it in a database table. For more information, see Saving the Concept History on page 28.

Table 2.3 Table View subtab - button descriptions

Button	Description
Differences	
Accept/Reject Diff 	Enable you to accept or reject differences that appear in the Differences panel. This panel shows mainly property changes.
Compare Changes 	Opens a Compare Versions window in which you can examine properties, asserted conditions, and disjoints for both the original class the changed class.
View Annotations 	Opens a Change annotations window showing the following change information: <ul style="list-style-type: none"> • Action: Type of change • Description: Nature of the change • Author: Person who made the change • Created: Date and time that the change was made

Table 2.3 Table View subtab - button descriptions (Continued)

Tree View Subtab

The **Tree View** subtab ([Figure 2.3](#) on page 20) is the prominent subtab immediately after the comparison process finishes running. This subtab shows the two compared ontologies in a single hierarchy.

The **Tree View** subtab displays three panels:

- The **Class Browser** panel displays the combined hierarchy. The number appearing just above the hierarchy displays the total number of changed classes since the previous baseline. Changed classes are shown in bold black text, and newly added classes are shown in blue, underlined text ([Figure 2.5](#)). Deleted classes are shown in red, strikethrough text.

Number of changed classes

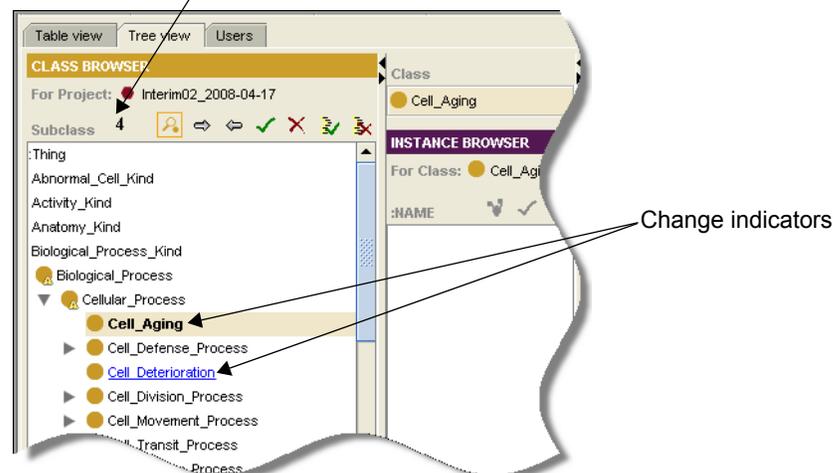


Figure 2.5 Class Browser - change indicators

- The **Instance Browser** panel shows the class that is currently selected in the Class Browser. Using the buttons on this panel, you can view references to an instance, accept an instance, or reject an instance.
- The **Class Editor** panel enables you to edit the selected instance.

To control the viewing space on the **Tree View** subtab, use the **Collapse** and **Expand** buttons at the edge of each panel (*Figure 2.6*).

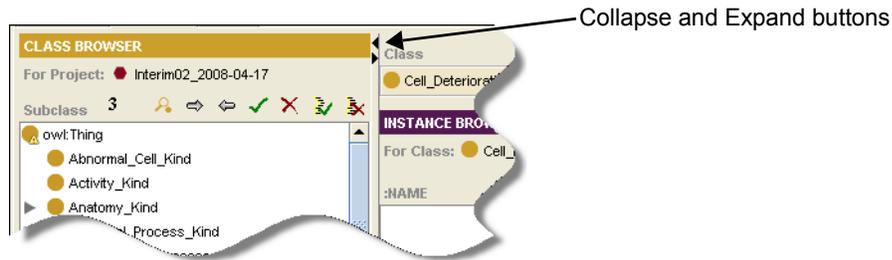


Figure 2.6 Collapse and Expand buttons

Table 2.4 describes the buttons used in each of the **Tree View** subtab panels.

Button	Description
Class Browser	
View Class 	Opens the Class Editor in a new window where you can perform all of the functions of the Class Editor panel (view, edit, and add notes to the selected class).
Next/Previous Change 	Used to browse forward and backward through changes in the ontology, one change at a time.
Accept/Reject Change 	Used to accept or reject the change that is currently selected in the class hierarchy.
Accept/Reject All Changes 	Used to accept or reject <i>all</i> changes shown for the current comparison.
Instance Browser	
View References to Instance 	Opens a window that shows references (frames, slots, and facets) to the selected instance.
Accept/Reject Instance Change 	Used to accept or reject a change to the selected instance.
Class Editor	
Show/Hide, Create, or Delete Class Notes 	Enables you to create class notes, show and hide them, and delete them.
Accept/Reject Diff 	Enable you to accept or reject differences that appear in the Differences panel. This panel shows mainly property changes.

Table 2.4 Tree View subtab - button descriptions for each panel

Button	Description
Compare Changes 	Opens a Compare Versions window in which you can examine properties, asserted conditions, and disjoints for both the original class the changed class.
View Annotations 	Opens a Change annotations window showing the following change information: <ul style="list-style-type: none"> • Action: Type of change • Description: Nature of the change • Author: Person who made the change • Created: Date and time that the change was made

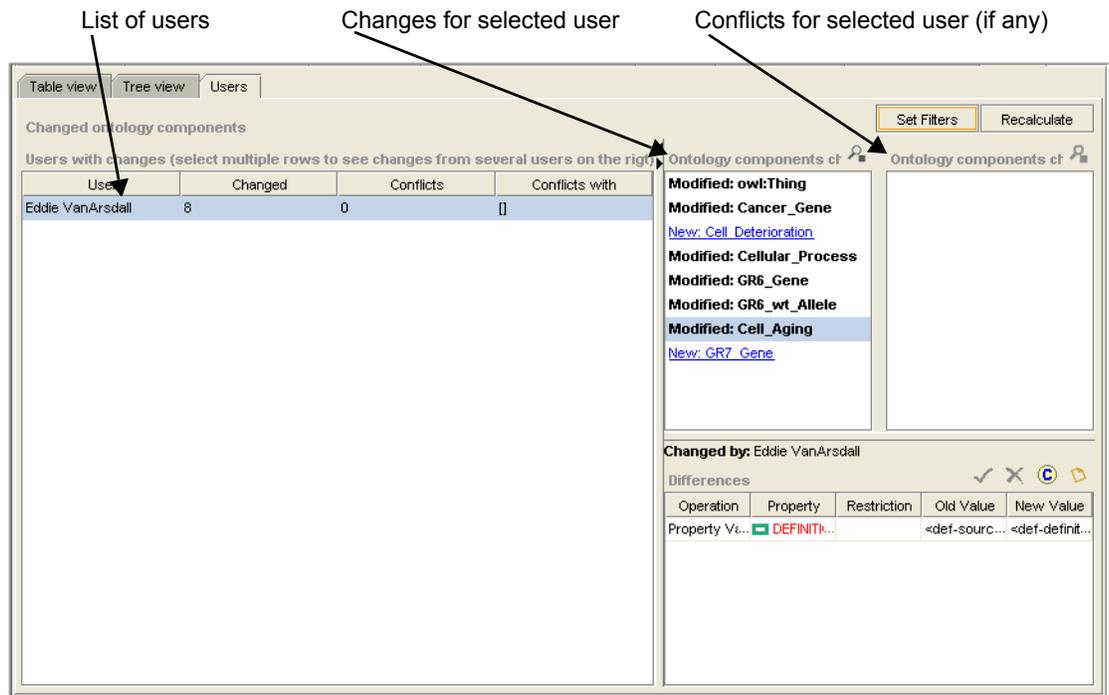
Table 2.4 Tree View subtab - button descriptions for each panel (Continued)

Users Subtab

The **Users** subtab lists each user who made changes to the ontology. The **Users with changes** panel on the left has four columns showing the following information:

- **User** shows the user's name.
- **Changed** shows the number of changes for each user.
- **Conflicts** shows the number of conflicts caused by a user's changes.
- **Conflicts with** shows the name of the user with conflicting changes.

The first of the two **Ontology component** panels on the right shows a list of changes made by a selected user. The second panel shows conflicts (if any) between the selected user's changes and another user's changes.



The screenshot shows the 'Users' subtab interface. It features a 'Table view' and 'Tree view' tab at the top left. Below the tabs is a 'Users with changes' table with the following data:

User	Changed	Conflicts	Conflicts with
Eddie VanArsdall	8	0	[]

Arrows point from labels to specific parts of the interface: 'List of users' points to the table, 'Changes for selected user' points to the 'Ontology components' panel, and 'Conflicts for selected user (if any)' points to the 'Differences' panel. The 'Ontology components' panel shows a list of changes for 'Eddie VanArsdall', including 'Modified: owl:Thing', 'Modified: Cancer_Gene', 'New: Cell Deterioration', 'Modified: Cellular_Process', 'Modified: GR6_Gene', 'Modified: GR6_wt_Allele', 'Modified: Cell_Aging', and 'New: GR7_Gene'. The 'Differences' panel shows a comparison of 'Property Vs...' with 'DEFINITI...' and 'Old Value' vs 'New Value'.

Figure 2.7 Users subtab

Table 2.5 describes the buttons used on the **Users View** subtab.

Button	Description
Set Filters 	Opens a window used to control what the panel displays: classes, properties, individuals, and anonymous ontology components.
Recalculate 	Updates the total number of changes. This is useful when you have accepted individual changes and want to update the total.
Ontology components panel	
View Changed Ontology Component 	Opens a window showing annotations for the item that is currently selected in either of the Ontology component panes. Select an item, then click this button to view its annotations.
Differences panel	
Accept/Reject Diff 	Enable you to accept or reject differences that appear in the Differences panel. This panel shows mainly property changes.
Compare Changes 	Opens a Compare Versions window in which you can examine properties, asserted conditions, and disjoints for both the original class the changed class.
View Annotations 	Opens a Change annotations window showing the following change information: <ul style="list-style-type: none"> • Action: Type of change • Description: Nature of the change • Author: Person who made the change • Created: Date and time that the change was made

Table 2.5 Users subtab - Button descriptions

Accepting and Rejecting Changes

Although you can accept or reject changes from various panels in the three **Prompt** subtabs, the following instructions focus on the toolbar found on the **Tree View** subtab.

Accepting a Single Change

To accept a single change, follow these steps:

1. Select the **Tree View** tab.
2. In the **Class Browser**, navigate to the change that you want to accept.

Tip: To identify changes in the hierarchy, look for a small triangular symbol with an exclamation mark inside it.
3. Select the change.
4. Using the toolbar at the top of the Class Browser, click the **Accept** button .

The name of the changed class no longer appears in bold or underlined text.

Accepting All Changes

When you run the **Accept All** command, you accept all changes identified in the current comparison. If you have already accepted or rejected individual changes, running this command accepts all of the *remaining* changes.

To accept all changes, follow these steps:

1. Select the **Tree View** tab.
2. Using the toolbar at the top of the Class Browser, click the **Accept All** button .

The names of the changed classes no longer appear in bold or underlined text.

Rejecting a Single Change

To reject a single change, follow these steps:

1. Select the **Tree View** tab.
2. In the **Class Browser**, navigate to the change that you want to reject.

Tip: To identify changes in the hierarchy, look for a small triangular symbol with an exclamation mark inside it.
3. Select the change.
4. Using the toolbar at the top of the Class Browser, click the **Reject** button .

The name of the changed class no longer appears in bold or underlined text.

Rejecting All Changes

When you run the **Reject All** command, you reject all changes identified in the current comparison. If you have already accepted or rejected individual changes, running this command accepts all of the *remaining* changes.

To reject all changes, follow these steps:

1. Select the **Tree View** tab.
2. Using the toolbar at the top of the Class Browser, click the **Reject All** button



The names of the changed classes no longer appear in bold or underlined text.

Saving the Concept History

The *concept history* provides a record of the edit actions (creation, merge, split, modification, and retirement) performed on a concept throughout its inclusion in the terminology. The history includes the dates on which each edit action occurred.

The concept history is saved in a database format and is intended for use in the publication of the terminology to an online server such as LexBig.

To save the concept history, follow these steps:

1. Click the **Table View** subtab.
2. In the **Image table** panel (upper panel), locate the toolbar on the far right.

Note: Before you complete the next step, note that there are two **Save** buttons in the toolbar. Although the correct button is on the far right, it could be moved in future releases. To ensure that you use the correct button, hover the mouse pointer over the button and make sure that the popup tool tip reads *Save in NCI DB Format*.

3. Click the second of the two **Save** buttons—the one on the far right.

The concept history is saved to a database table.

Disabling the Prompt Tab

After completing a database comparison, disable the **Prompt** tab to ensure that it does not appear in the main Protégé project.

To disable the **Prompt** tab, follow these steps:

1. Select the following menu command: **Project > Configure...**

The **Configure File** window opens.

2. In the **Visible** column, clear the check box to the left of the **Prompt** widget.
3. Click **OK** to close the **Configure File** window.

The **Prompt** tab is no longer visible.

Preparatory Tasks for the Next Comparison

After you have run a comparison, reviewed the results, and accepted or rejected changes, perform the following tasks to prepare for the next comparison.

- [Changing the Baseline Version Number for the Next Prompt Build](#) on this page
- [Exporting a Project File](#) on page 30
- [Deleting Extraneous Information](#) on page 31

Naming a Baseline Version

When you export the current baseline file for use during the next comparison, you use a specified naming convention for the file. You also use a specified convention for the `owl:versionInfo` property shown on the Protégé **Metadata** tab.

Example: Naming the Exported File

Today is Friday, April 4, 2008, and you have just run a comparison for the BiomedGT project. This is your first comparison in April 2008.

When you export the project file, the file name uses the following string:

- project name: BiomedGT
- an underscore: `_`
- the year (`yy`), month (`mm`), and day (`dd`) of the *current* Prompt build: `yymmdd`
- a lowercase letter representing the *current* Prompt build:
a for the first, b for the second, c for the third, etc.)

In this example, you have just run the first build of the month. Thus, when you export the file, you give it the following name:

```
BiomedGT_080404a
```

Example: Specifying the owl:versionInfo Property Value

In addition to exporting a file for the *current* baseline, you also need to express the value of the *next* Prompt build using the `owl:versionInfo` property displayed on the **Metadata** tab.

The value of the `owl:versionInfo` property follows this format:

- year of the current Prompt build: `yy`
- a period: `.`
- month of the current Prompt build: `mm`
- a lowercase letter representing the *next* Prompt build; using the first example, if the current build is letter a, then the next build will be letter b.

Continuing with the above example, if you just ran your first comparison of the month on April 4, 2008, then the `owl:versionInfo` property will show a letter b to represent the *second* build of the month.

```
08.04b
```

Changing the Baseline Version Number for the Next Prompt Build

Prerequisite reading: [Naming a Baseline Version](#) on page 29

For baseline version tracking, change the value of the `owl:versionInfo` property on the Protégé **Metadata** tab to show the current year, the current month, and the baseline version of the *next* Prompt build.

To change the value of the `owl:versionInfo` property, follow these steps:

1. Click the **Metadata** tab.

The panel on the right shows two columns: **Property** and **Value**. The first listed property, `owl:versionInfo`, shows the baseline version number.

2. Double-click the value column for `owl:versionInfo`.
3. Change the value to reflect the next Prompt build.

For example, if you just ran your second Prompt build on May 11, 2008, and plan to run a third build next week, the value would `yy.mmcc`, or `08.05c`.

4. Click outside of the **Value** column to deselect the property.

Exporting a Project File

Prerequisite reading: [Naming a Baseline Version](#) on page 29

At the end of each baseline comparison cycle, you export a copy of the current master project file by converting it to an OWL file. You then use the exported file as a baseline for comparison during the next cycle. This enables you to identify changes that have been made between cycles and accept or reject them.

To export the project file, follow these steps:

1. Select the following menu command: **File > Convert Project to Format**

The **Select Format** window opens.

2. Select **OWL / RDF Files** from the list, then click **OK**.

The **OWL / RDF Files** window opens. In the **OWL file name or URL** field, note that the OWL file is given the same name as the project (`.pproj`) file.

3. Ensure that the name shown in the **OWL file name or URL** field is in the following format:

```
{project name}_yymmdda.owl
```

where the letter *a* represents the first build of the month, *b* represents the second, *c* represents the third, and so on.

4. If the file name does not follow the established convention, change it in the **OWL file name or URL** field.
5. Leave the Language set to **RDF/XML-ABBREV**.
6. Click **OK**.

Deleting Extraneous Information

During a baseline comparison, Prompt identifies changes by using information stored both in the Protégé master project file and in other external files on the host computer. Once the comparison has run and you have addressed all of the changes, the project information and the external files must be deleted in preparation for the next comparison.

Change Ontology

When you first log in to Protégé and select a project, you may notice a project named *Annotation_BiomedGT (do not edit)*. This is the *change ontology* that tracks all changes and shows who made them. While using Protégé, you can view the current change ontology using the **Changes** tab.

Prompt uses information in the change ontology to determine changes in the current baseline. To prepare for the next round of editing, you must clear this information so that Protégé will start a new version. Prompt will then use the new version during the next comparison.

Annotation Ontology

Editors' changes are also stored in an external file called an *annotation ontology* (also a .pprj file). This file serves as an audit trail of changes as they take place. During a comparison, Prompt uses this file to determine changes between the current project file and the most recent baseline file.

The annotation ontology file is accompanied by two other files with identical names but different extensions (.rdf and .rdfs). The file names are prefixed with the word *annotation*, an underscore, and the project name.

For example, for a project called *BiomedGT*, you might see the following files:

- annotation_BiomedGT.pprj
- annotation_BiomedGT.rdf
- annotation_BiomedGT.rdfs

Once you review the changes that are tracked in the current annotation ontology, you can delete these files to prepare for the next round of editing. Before deleting these files, you should exit Protégé without saving the master project.

“Accepts” and Journal File

Other files that must be deleted include a file called *Thesaurus.pjrn*, also known as a *journal* file, and any files that match the project name but have the word “accepts” appended to the name. Usually there are three of these files, with the extensions .pprj, .pont, and .pins.

For a project called BiomedGT, you might see the following files:

- BiomedGT-accepts.pprj
- BiomedGT-accepts.pont
- BiomedGT-accepts.pins

You should also delete these files while Protégé is closed.

Deleting Both Ontologies

To delete the change ontology and the specified external files, follow these steps:

1. Select the following menu command: **File > Exit Protégé**
A prompt appears, asking if you want to save changes to the current project.

2. Click **No**.

Protégé exits without saving the current project.

Note: Except for the journal file with the .pjrn extension, the files listed in the next step use the project name *BiomedGT*. If your project name is different, delete the files that match your project name but have the same extensions.

3. Using the file manager, delete the following files:

- annotation_BiomedGT.pprj
- annotation_BiomedGT.rdf
- annotation_BiomedGT.rdfs
- BiomedGT-accepts.pprj
- BiomedGT-accepts.pont
- BiomedGT-accepts.pins
- Thesaurus.pjrn

4. Restart Protégé.

5. After the project loads, select the following menu command:
File > Save Project.

This creates a new change ontology file.

6. Exit Protégé.

Final Tasks

The following procedure corresponds to steps 17 through 20 in *Table 2.1* (page 16).

To restart the servers and log out of the Protégé host computer, follow these steps:

1. Restart or start the Explanation server, depending on whether you previously shut it down.

Restart the Explanation server (if not previously shut down):

- a. Type the following command at the command prompt:

```
cd /usr/local/protege/Explanation.Server-x.y.z
```

- b. Press ENTER.
- c. Type the following command:

```
./restart_explanation_server.sh
```

- d. Press ENTER.

Start the Explanation server (if previously shut down):

- a. Type the following command at the command prompt

```
cd /usr/local/protege/Explanation.Server-x.y.z
```

- b. Press ENTER.
- c. Type the following command:

```
./start_explanation_server.sh -p 8090 --url jdbc:mysql//  
cbiows501.nci.nih.gov:3610/protege -t ThesaurusBaseline -U  
protege -P
```

- d. Press ENTER.

2. Start the Protégé server:

- a. Type the following command at the command prompt:

```
cd /usr/local/protege/Protege.Server-x.y.z
```

- b. Press ENTER.
- c. Type the following command:

```
./run_protege_server.sh
```

- d. Press ENTER.

3. Verify that the Protégé server is running:

- a. Type the following command at the command prompt:

```
cd /usr/local/protege/Protege.Server-x.y.z
```

- b. Press ENTER.
- c. Type the following command:

```
tail -f console.txt
```

- d. Press ENTER.

4. Verify that the **Explanations** tab is communicating with the Explanation server:
 - a. Re-open the Protégé client.
 - b. Click the **Explanations** tab.
 - c. Using the Class Browser on the left, select a class.
The **Named Superclass** and **Named Subclasses** panels (in the center of the tab) show the selected class and its children.
 - d. Select a named subclass.
An explanation appears in the far right panel. If no explanation appears, or if an error message appears, then the communication failed. If this happens

Note: A communication failure between the **Explanations** tab and the Explanation server is considered an serious error. The problem should be quickly reported to the EVS Operations team by phone instead of through GForge.
5. Log off of the host computer using Reflection X.

From Here...

- [Classification Procedures](#) on page 35

CHAPTER 3

CLASSIFICATION PROCEDURES

This chapter explains how to classify the NCI Protégé database, accept or reject proposed changes, and prepare and save a classification report for distribution.

Topics in this chapter:

- [Overview](#) on this page
- [Preliminary Tasks](#) on page 36
- [Classification Tasks](#) on page 38
- [Final Tasks](#) on page 43

Overview

To perform a periodic inferencing analysis of the NCI Protégé database, you run an automated classification process using a tool called a *reasoner*. The reasoner examines the relationships in the class hierarchy, computes the subclass relations between every named class, and determines whether possible modeling errors exist.

About the Classification Process

[Table 3.1](#) summarizes the process stages for classification.

Stage	Action	Responsibility
Preliminary Tasks on page 36		
1.	Send e-mail or chat to alert modelers that the NCI Protégé database will be shut down and that they need to log out.	Manager
2.	Log out of the database.	Modeler
3.	Log into the Protégé host computer.	Manager
4.	Shut down the Protégé server.	Manager

Table 3.1 Classification process

Stage	Action	Responsibility
5.	Start the remote Protégé client and connect to the BiomedGT project.	Manager
<i>Classification Tasks</i> on page 38		
6.	Configure the remote Protégé client to display the OWL Classes tab.	Manager
7.	Start the reasoner.	Manager
8.	Review the results, accepting or rejecting retreeing suggestions according to established guidelines.	Manager
9.	Using the previous classification report reviewed and discussed by the modelers, accept or reject items as determined.	Manager
10.	Save the current classification results as an external text-delimited file, then open the file in Microsoft Excel and save it as an Excel workbook for circulation among the team.	Manager
11.	Exit the Protégé project without saving it.	Manager
<i>Final Tasks</i> on page 43		
12.	Restart the Protégé server and verify that it is running.	Manager
13.	(Optional) Restart the Explanation server to make sure that it is synchronized with the database.	Manager
14.	Log off of the Protégé host computer.	Manager

Table 3.1 Classification process (Continued)

About the Pellet Reasoner

NCI Protégé uses the Pellet reasoner, an open-source tool developed and commercially supported by Clark & Parsia LLC (<http://clarkparsia.com/>). As stated on the Pellet Web site, “Pellet supports the full expressivity of OWL DL, including reasoning about nominals (enumerated classes).”

To learn more about the Pellet reasoner, visit <http://pellet.owldl.com/>.

Preliminary Tasks

The following procedures correspond to steps 1 through 5 in *Table 3.1* (page 35).

- *Notifying Protégé Users of Database Shutdown* on this page
- *Starting the Remote Protégé Client* on page 37

Notifying Protégé Users of Database Shutdown

Prior to performing a classification, follow these steps:

1. Using e-mail or chat, announce that the Protégé database will be shut down.
2. Log out of the Protégé client on your own machine.

Starting the Remote Protégé Client

Note: In the following instructions, the current NCI Protégé directory is represented as `Protege.Server-x.y.z`. When typing commands, substitute the directory name with the current Protégé version number (**Example:** `Protégé.Server-1.2.3`).

To start the Protégé client on the remote server, follow these steps:

1. Log in to the Protégé host computer:
 - a. Open Reflection X.
 - b. Select the **direct** connection method.
 - c. Enter the host machine name.

Tip: If you have previously entered the host machine name, you can select it from the drop-down list.
 - d. Enter your user name and password.
2. Open a Terminal window.
3. Shut down the Protégé server:
 - a. Type the following command at the command prompt:

```
cd /usr/local/protege/Protege.Server-x.y.z
```
 - b. Press ENTER.
 - c. Type the following command:

```
./shutdown_protege_server.sh localhost:2633
```
 - d. Press ENTER.
4. Start the remote Protégé client:
 - a. Open a second Terminal window.
 - b. Type the following command:

```
cd /usr/local/protege/Protege.Client-x.y.z
```
 - c. Press ENTER.
 - d. Type the following command:

```
./run_protege.sh
```
 - e. Press ENTER.
 - f. When prompted to select a project, select **BiomedGT**.

Classification Tasks

The following procedures correspond to steps 6 through 11 in *Table 3.1* (page 35).

- *Enabling the OWL Classes Tab* on this page
- *Starting the Reasoner* on page 39
- *Asserting (Accepting or Rejecting) Current Classification Results* on page 39
- *Saving the Current Classification Results as a Report* on page 42

Enabling the OWL Classes Tab

After running the reasoner, you can view the results using the **OWL Classes** tab. This tab is not enabled by default.

To enable the **OWL Classes** tab, follow these steps:

1. Select the following menu command: **Project > Configure...**
The **Configure File** window opens.
2. In the **Visible** column, check the box to the left of the **OwlClassesTab** widget.
3. Click **OK** to close the **Configure File** window.

The **OWL Classes** tab now appears in the main Protégé window. The tab is positioned on the far right (*Figure 3.1*).

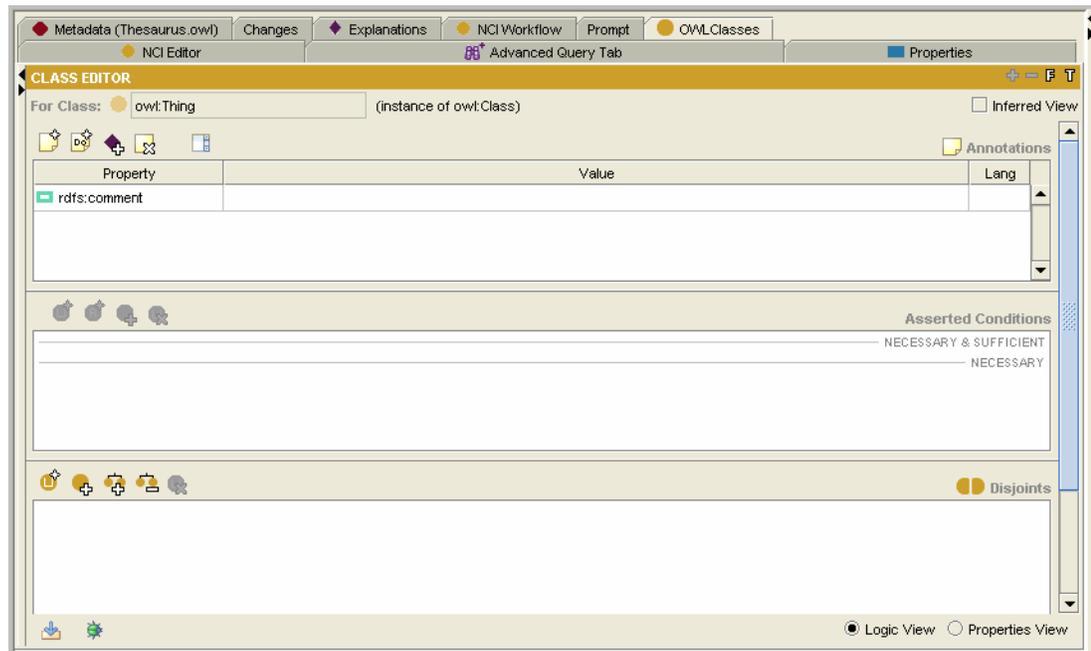


Figure 3.1 Owl Classes tab

Starting the Reasoner

Prerequisite task: *Enabling the OWL Classes Tab* on page 38

To start the reasoner from the **OWL Classes** tab, select the following menu command:

OWL > Classify taxonomy

A status window opens. This window monitors the status of the classification process. When the **Task Complete** message shows (*Figure 3.2*), click **OK** to close the window.

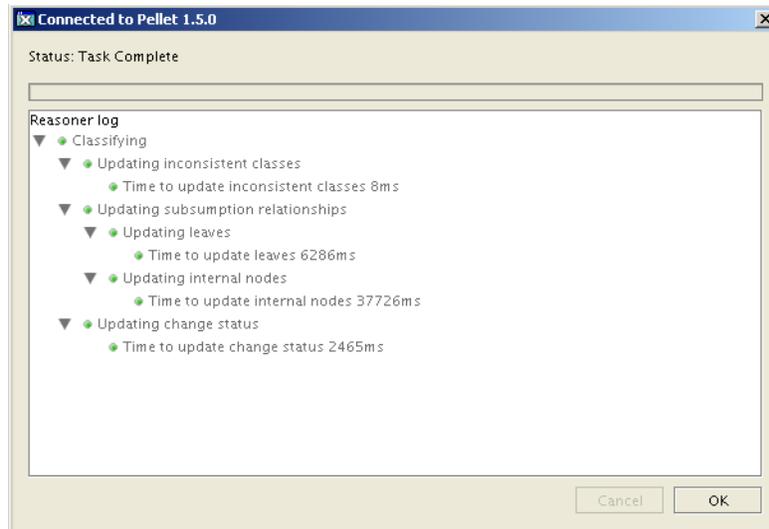


Figure 3.2 Window showing status during classification

Asserting (Accepting or Rejecting) Current Classification Results

After the reasoner runs, click the **OWL Classes** tab and examine the layout. Note the following aspects of the tab panels (illustrated in *Figure 3.3* on page 40):

- The two **Subclass Explorer** panels each represent a different view of the class hierarchy:
 - The left panel displays the *Asserted Hierarchy*. This is the hierarchy resulting from current modeling.
 - The middle panel displays the *Inferred Hierarchy*. This is the hierarchy as established by the reasoner during classification.
- The **Class Editor** panel is a fully functional editor, enabling you to view and edit properties for a selected class or subclass.
- The **Classification Results** panel at the bottom of the tab shows class names on the left and changed direct subclasses on the right.

Scenario 1: Accepting a Suggestion

In *Figure 3.3*, the *Bendamustine* class is selected in the class results. The reasoner has suggested that the redundant parent class *Antineoplastic Alkylating Agent* (shown in the **Class Editor** panel) be removed.

To accept the selected change, click the **Assert Selected Change** button  on the left of the **Classification Results** panel (*Figure 3.3*). The button icon resembles a wrench.

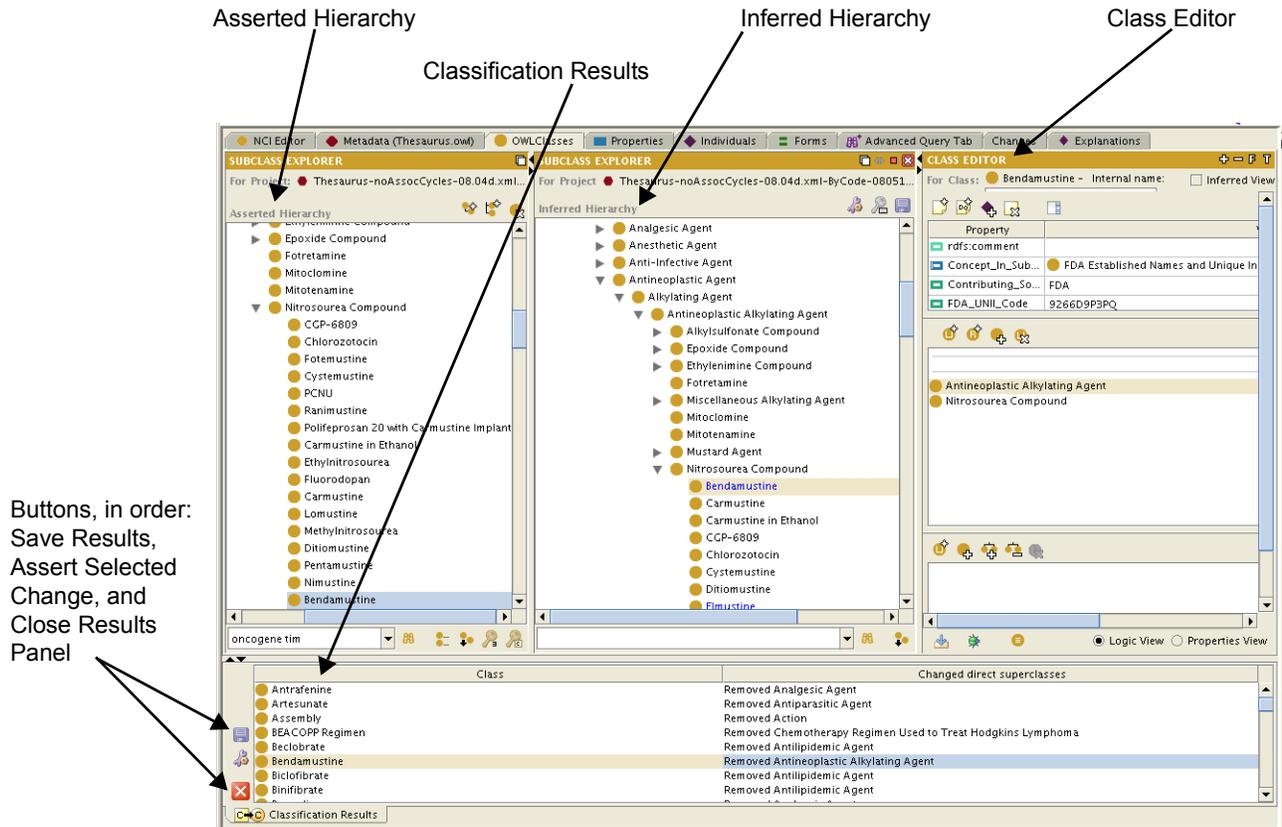


Figure 3.3 OWL Classes tab with classification results

Scenario 2: Rejecting a Suggestion

In [Figure 3.4](#), the reasoner has suggested that Complement Components 3 and 5 be re-treed as children of Complement Component 4. This suggestion is probably not valid and should alert the modeler that either (1) the DL model is incorrect and needs to be reconsidered, or (2) the DL model has been incorrectly applied and the modeler needs to modify one or more of the concepts in question.

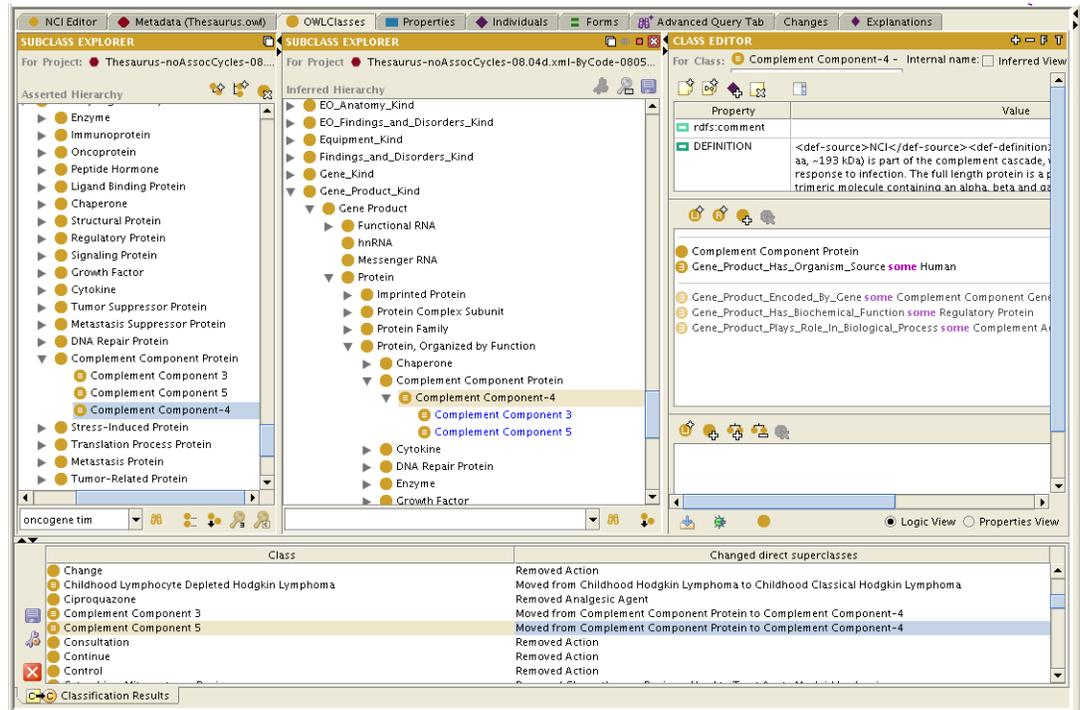


Figure 3.4 Classification Results panel with suggestions for Components 3 and 5

Note: The Classification Results panel does not include a button or any other mechanism for rejecting a suggestion. You reject suggestions by not explicitly accepting them. For the current scenario, ignore the suggestion and make a note to follow up on the three sibling classes.

Asserting (Accepting or Rejecting) Results from the Previous Classification

After asserting results from the *current* classification, review the report from the previous classification. Assert any outstanding items that have been resolved through discussion with the modelers.

For more information about creating a report after each classification session, see [Saving the Current Classification Results as a Report](#) on page 42.

Saving the Current Classification Results as a Report

Once you have identified any modeling issues that you need to discuss with the team, you can create a report for circulation.

To create a report showing the current classification results, follow these steps:

1. In the Classification Results (bottom) panel, click the **Save Classification Results** button on the left (*Figure 3.5*).

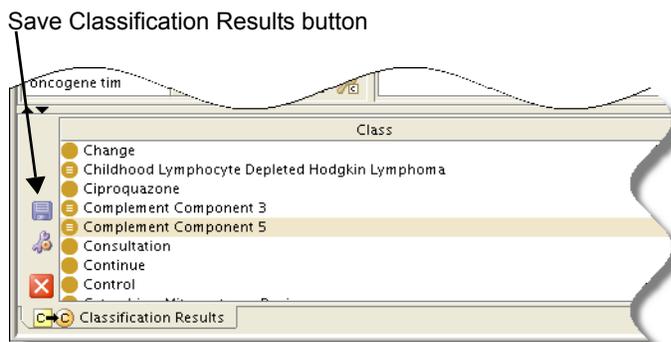


Figure 3.5 Save Classification Results button

2. Save the file to a specified directory, noting where you saved it. This step creates a text file that is delimited by equal signs (=).
3. Open Microsoft Excel.
4. Select **File > Open**, then open the file that you saved in step 2. The Text Import Wizard launches.
5. To set up the workbook, use the following Wizard settings:
 - a. On the Step 1 of 3 screen, select **Delimited** as the file type, then click **Next**.
 - b. On the Step 2 of 3 screen,
 - Clear the **Tab** check box under the **Delimiters** section.
 - Select the **Other** check box.
 - Type an equal sign (=) in the box following the word *Other*.
 - Click **Next**.
 - c. On the Step 3 of 3 screen, click **Finish**.

The delimited data now appear in workbook columns.
6. Using **File > Save As**, save the file in Microsoft Excel (.xls) format.
7. Close Excel.

Closing Protégé Without Saving the Project

In an earlier procedure, you configured the open Protégé project to display the **OWL Classes** tab. Since this tab isn't part of the standard configuration, remember to exit Protégé without saving the project.

Final Tasks

The following procedure corresponds to steps 12 through 14 in *Table 3.1* (page 35).

To complete the classification process, follow these steps:

1. Start the Protégé server:

- a. Type the following command at the command prompt:

```
cd /usr/local/protege/Protege.Server-x.y.z
```

- b. Press ENTER.

- c. Type the following command:

```
./run_protege_server.sh
```

- d. Press ENTER.

2. Verify that the Protégé server is running:

- a. Type the following command:

```
tail -f console.txt
```

- b. Press ENTER.

Note: You can also verify that the Protégé server is running by re-opening the Protégé client and then closing it.

3. (Optional) Restart the Explanation server to make sure that it is synchronized with the database:

- a. Type the following command at the command prompt:

```
cd /usr/local/protege/Explanation.Server-x.y.z
```

- b. Press ENTER.

- c. Type the following command:

```
./restart_explanation_server.sh
```

- d. Press ENTER.

4. Log off of the host computer using Reflection X.

INDEX

A

- accepting
 - classification results 39 – 41
 - comparison results 27
- accepts and journal files 31
- annotation ontology, about 31
- asserting classification results 39 – 41
- assigning workflow item 12

B

- baseline version
 - changing number 30
 - naming 29
- BiomedGT overview 5 – 7

C

- change ontology, about 31
- changing baseline version number 30
- classification process
 - asserting results 39 – 41
 - overview 35 – 36
- comparing baseline files
 - See database comparison process
- concept history, saving 28
- contacting support 4

D

- database comparison process
 - about 15
 - accepting and rejecting results 27
 - deleting extraneous files 31 – 32
 - interpreting results 21
 - starting 19

E

- Explanation server
 - shutting down 17
 - starting and verifying 33
- exporting project file 30

I

- importing packages from wiki 12

J

- journal and accepts files 31

N

- naming baseline version 29
- NCI Workflow tab 10
- notifying users of shutdown 17, 36

O

- Owl Classes tab, enabling 38

P

- Pellet reasoner
 - about 36
 - starting 39
- project file, exporting 30
- Prompt plug-in, about 17
- Prompt tab
 - disabling 28
 - enabling 18
- Protégé server
 - shutting down 17
 - starting and verifying 33

R

- reasoner (Pellet)
 - about 36
 - starting 39
- reassigning workflow item 12
- rejecting
 - classification results 39 – 41
 - comparison results 27
- remote client, starting 17, 37
- results (classification)
 - accepting and rejecting 39 – 41
- results (comparison)

accepting and rejecting 27
interpreting 21

S

Semantic MediaWiki

See BiomedGT overview

starting remote client 17, 37

support, contacting 4

T

Table View subtab 21

Tree View subtab 23

U

users, notifying of shutdown 17, 36

Users subtab 25

W

workflow overview

BiomedGT to Protégé 7

internal NCI process 8 – 11

workflow procedures

assigning item 12

importing packages from wiki 12