caBench-to-Bedside Admin module v2.1

Administration Manual

Document Change History

Version Number	Date	Contributor	Description
V1.0	June 25, 2008	Washington University/Persistent Systems	Reviewed
V2.1	December 4, 2008	Washington University/Persistent Systems	Reviewed
V2.1	December 23,2008	Washington University/Persistent Systems	Reviewed by Documentation and Training workspace

This is a U.S. Government work.

December 24, 2008

Development Team(s)		
Development	***User's Guide***	***Program Management***
Chandrakant Talele[1]	Chandrakant Talele[1]	Dr. Rakesh Nagarajan[2]
Srinath Krishnamurthy[1]	Srikanth Adiga[1]	Srikanth Adiga[1]
Deepak Shingan[1]	Madhumita Shrikhande[1]	
Chetan Patil[1]	Madhurima Bhattacharjee[1]	
Hrishikesh Rajpathak[1]	Mukesh Sharma[2]	
Lalit Chand[1]		
Atul Jawale[1]		
[1]Persistent Systems Limited	[2]Washington University	

Contacts and Support	
Training contact	cab2b@bmi.wustl.edu
Customer support contact	cab2b@bmi.wustl.edu

Copyright and License Information

Copyright Notice Copyright 2008 School of Medicine, Washington University in St. Louis ("caBIG® Participant"). ca Bench-to-Bedside Admin Module was created with NCI funding and is part of the caBIG® initiative. The software subject to this notice and license includes both human readable source code form and machine readable, binary, object code form (the "caBIG® Software").

This caBIG® Software License (the "License") is between caBIG® Participant and You. "You (or "Your") shall mean a person or an entity, and all other entities that control, are controlled by, or are under common control with the entity. "Control" for purposes of this definition means (i) the direct or indirect power to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

License Provided that You agree to the conditions described below, caBIG® Participant grants you a non-exclusive, worldwide, perpetual, fully-paid-up, no-charge, irrevocable, transferable and royalty-free right and license in its rights in the caBIG® Software, including any copyright or patent rights therein, to (i) use, install, disclose, access, operate, execute, reproduce, copy, modify, translate, market, publicly display, publicly perform, and prepare derivative works of the caBIG® Software in any manner and for any purpose, and to have or permit others to do so; (ii) make, have made, use, practice, sell, and offer for sale, import, and/or otherwise dispose of caBIG® Software (or portions thereof); (iii) distribute and have distributed to and by third parties the caBIG® Software and any modifications and derivative works thereof; and (iv) sublicense the foregoing rights set out in (i), (ii) and (iii) to third parties, including the right to license such rights to further third parties. For sake of clarity, and not by way of limitation, caBIG® Participant shall have no right of accounting or right of payment from You or Your sublicensees for the rights granted under this License. This License is granted at no charge to You. Your downloading, copying, modifying, displaying, distributing or use of caBIG® Software constitutes acceptance of all of the terms and conditions of this Agreement. If you do not agree to such terms and conditions, you have no right to download, copy, modify, display, distribute or use the caBIG® Software.

1. Your redistributions of the source code for the caBIG® Software must retain the above copyright notice, this list of conditions and the disclaimer and limitation of liability of Article 6 below. Your redistributions in object code form must reproduce the above copyright notice, this list of conditions and the disclaimer of Article 6 in the documentation and/or other materials provided with the distribution, if any.

2. Your end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by School of Medicine, Washington University in St. Louis." If You do not include such end-user documentation, You shall include this acknowledgment in the caBIG® Software itself, wherever such third-party acknowledgments normally appear.

3. You may not use the names "School of Medicine, Washington University in St. Louis", "The National Cancer Institute", "NCI", "Cancer Bioinformatics Grid" or "caBIG®" to endorse or promote products derived from this caBIG® Software. This License does not authorize You to use any trademarks, service marks, trade names,

logos or product names of either caBIG® Participant, NCI or caBIG®, except as required to comply with the terms of this License.

4. For sake of clarity, and not by way of limitation, You may incorporate this caBIG® Software into Your proprietary programs and into any third party proprietary programs. However, if You incorporate the caBIG® Software into third party proprietary programs, You agree that You are solely responsible for obtaining any permission from such third parties required to incorporate the caBIG® Software into such third party proprietary programs and for informing Your sublicensees, including without limitation Your end-users, of their obligation to secure any required permissions from such third parties before incorporating the caBIG® Software into such third party proprietary software programs. In the event that You fail to obtain such permissions, You agree to indemnify caBIG® Participant for any claims against caBIG® Participant by such third parties, except to the extent prohibited by law, resulting from Your failure to obtain such permissions.

5. For sake of clarity, and not by way of limitation, You may add Your own copyright statement to Your modifications and to the derivative works, and You may provide additional or different license terms and conditions in Your sublicenses of modifications of the caBIG® Software, or any derivative works of the caBIG® Software as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

6. THIS caBIG® SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES (INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE) ARE DISCLAIMED. IN NO EVENT SHALL THE SCHOOL OF MEDICINE, WASHINGTON UNIVERSITY IN ST. LOUIS OR ITS AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS caBIG® SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents

About This Guide **1**
Purpose 1
Typical User 1
Topics Covered 1
Text Conventions Used 2

Chapter 1 Introduction **3**
Introduction to the caB2B Admin Module 3
Overview of the manual 4
Organization of the manual 4
Starting the caB2B Admin Module 5
Document Text Conventions 7

Chapter 2 Load Models **8**
Introduction 8
What is a Model? 8
What is Metadata Repository? 8
Loading Models 9

Chapter 3 Service Instances **12**
Introduction to the Service Instances 12
Configuring the Service Instances 12

Chapter 4 Create Category **16**
What is a Category? 16
What is a Path? 17
What is an Ambiguity Resolver? 17
How to create a Category? 18
Advanced Search 24

Chapter 5 Curate Paths **26**
Introduction to the Curated Paths 26
Creating Curated Paths 26
Use of the Curated Path 30

Chapter 6 Inter Model Joins **35**
Need for Inter Model Joins 35
Introduction to Inter Model Joins 35
Creating an Inter model Join 36
Using Inter model Join 39

Index **41**

About This Guide

This section introduces you to the cancer Bench-To-Bedside (caB2B) Admin module. Topics in this section include

- Purpose on page 1
- Typical User on page 1
- Topics Covered on page 1
- Text Conventions Used on page 2

Purpose

This manual provides the details of how to use caBench-To-Bedside (caB2B) Admin module to set up, configure and manage the core caBench-To-Bedside (caB2B) application.

All the chapters in this admin manual begin with an introductory statement and are followed by a list of topics included in that chapter.

Typical User

This guide is designed for the Administrators of caBench-To-Bedside (caB2B) application, who configure and manage the caB2B application. The administrator should be familiar with domain models. He should have an understanding of what are the classes and attributes present in a domain (UML) model. Administrator should have knowledge of bioinformatics so that he can understand the various models that are available for query within caB2Band can configure the client. This would make it easier for an novice end user who does not has any clue about what a model is.Administrator should have a basic knowledge of computers which would help him to use the admin module which is a web based application. No particular computer skill is required.

Topics Covered

- "Chapter 1: Introduction"
- "Chapter 2: Load Models"
- "Chapter 3: Service Instances"
- "Chapter 4: Create Category"
- "Chapter 5: Curate Paths"
- "Chapter 6: Inter Model Joins"
- Index

Text Conventions Used

This section explains conventions that are 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 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.
Italics	Highlights references to other documents, sections, figures, and tables.	See *Figure 4.5*.
Italic boldface monospace type	Represents text that you type.	In the **New Subset** text box, enter ***Proprietary Proteins***.
Note:	Highlights information of particular importance.	**Note:** This concept is used throughout this document.
{ }	Surrounds replaceable items.	Replace {last name, first name} with the Principal Investigator's name.

Chapter 1 Introduction

This chapter explains the scope and function of the cancer Bench-To-Bedside (caB2B) Admin Module, a component of the core caB2B application. It explains how the caB2B Admin Module helps in the configuring and setting up and configuring of the core caB2B application.

This chapter contains the following topics:

- Introduction
- Overview of the manual
- Organization of the manual
- Starting the caB2B Admin Module
- Document Text Conventions

Introduction to the caB2B Admin Module

The **cancer BenchTo-Bedside (caB2B)** application is used to query and retrieve useful data and information from the various data services that are **caGrid** enabled. The **caB2B Admin Module** (or simply, admin module) is a component of the core caB2B application, which allows the Administrator of the caB2B application to configure and define various properties of the application. Before launching the caB2B client the administrator has to configure caB2B for the users, where the administrator can load models from caDSR as per the requirement of his organization, create categories which are useful for the users,curate the best possible path between the classes and create intermodel joins between two classes. These customized categories and curated paths make the application easier for the end users.

The admin module is a web-based application used by the administrator to carry out the following tasks:

- **Define the categories to be queried**

  Using the admin module, the Administrator can define the categories by combining entities from a UML model, to make the task of querying easier for the end-user.

- **Load new models from the caDSR**

  The admin module can load the models present in the cancer Data Standards Repository (caDSR). Once the model is loaded, it is possible to query the model using the caB2B client.

- **Define the service instances to be used**

  Using the admin module, the Administrator can specify the instances to be queried for a given model. The Administrator can specify one or more instances of the services available in caB2B. These can be queried by dynamically discovering services running on the caGrid.

- **Curate path between categories**

  Using the admin module, the administrator can curate paths between various domain objects. By default, there are several paths available between any two categories. From these paths, the administrator can select only those paths that are useful for the end-user.

- **Define Inter model joins**

  The Administrator can make two models interoperable by defining a bridge between the two different models. The domain objects with the common data element concept can be connected with each other, providing a connection between different models.

Overview of the manual

To know how to use caB2B Admin Module to:

- Load metadata into the repository, refer to "Chapter 2 Load Models".
- Determine the service instances to query, refer to "Chapter 3 Service Instances".
- Create the categories, refer to "Chapter 4 Create Category".
- Curate paths to help the end-user to build queries, refer to "Chapter 5 Curate Paths".
- Create the Inter Model Joins to bridge different models, refer to "Chapter 6 Inter Model Joins".
- To start with the Admin module administrator needs to install it, refer to Admin installation guide to install Admininstrative module.
- Admin module is used to configure caB2B client, to start with caB2B client refer to caB2B user manual and caB2B installation guide.

Organization of the manual

The following table summarizes the chapters in this manual:

Chapter in caB2B Admin Module	Chapter Contents
Introduction	Introduction to the caB2B Admin Module Overview of the manual Organization of the manual Starting the caB2B Admin Module Document Text Conventions

Load Models	Introduction Loading Models
Service Instance	Introduction to the Service Instances Configuring the Service Instances
Create Categories	What is a Category? What is a Path? How to create a Category?
Curate Paths	Introduction to the Curated Paths Creating Curated Paths Use of the Curated Path
Inter model Joins	Need for Inter Model Joins Introduction to Inter Model Joins Creating an Inter model Join Using Inter model Join

Starting the caB2B Admin Module

To access the Admin module, go to the application URL. The system navigates to the Login Page of the Admin module, as shown below in Figure 1.1.

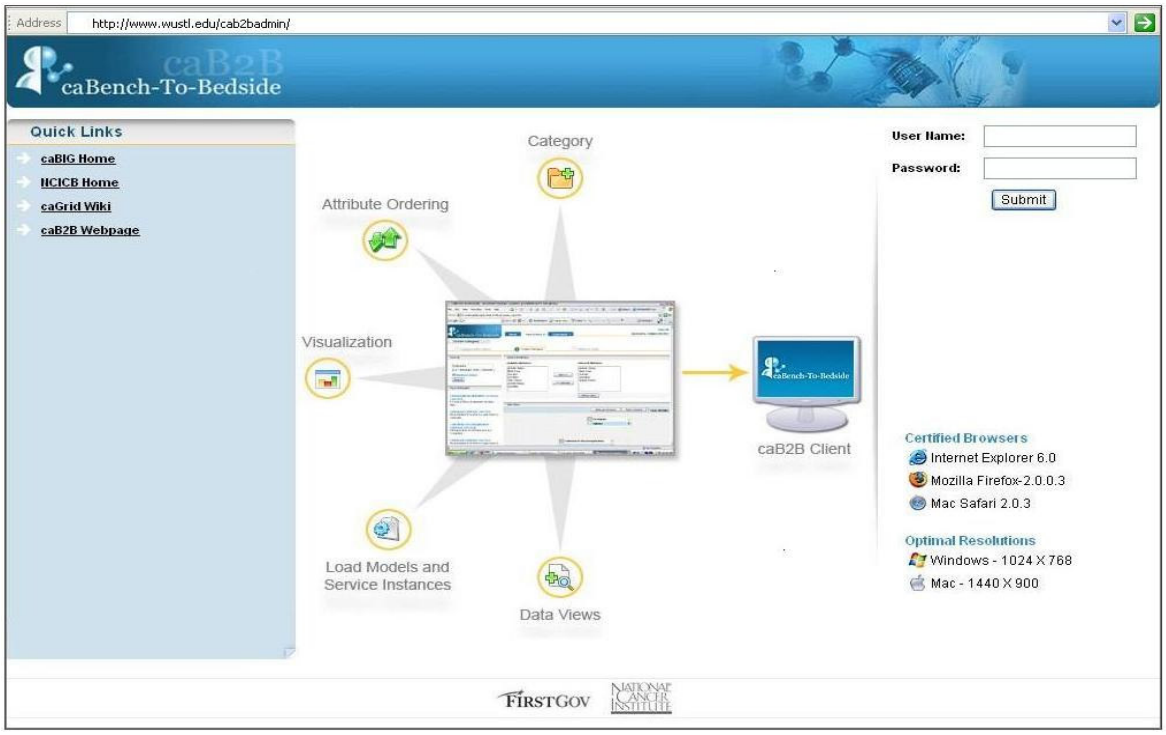

Figure 1.1. caB2B Admin Module Login Page

You require a valid **User Name** and **Password** to log into the admin module. Refer to the Installation Guide for the default admin username and password. Once you log into the admin module, you see the admin module's Home Page shown below in Figure 1.2.

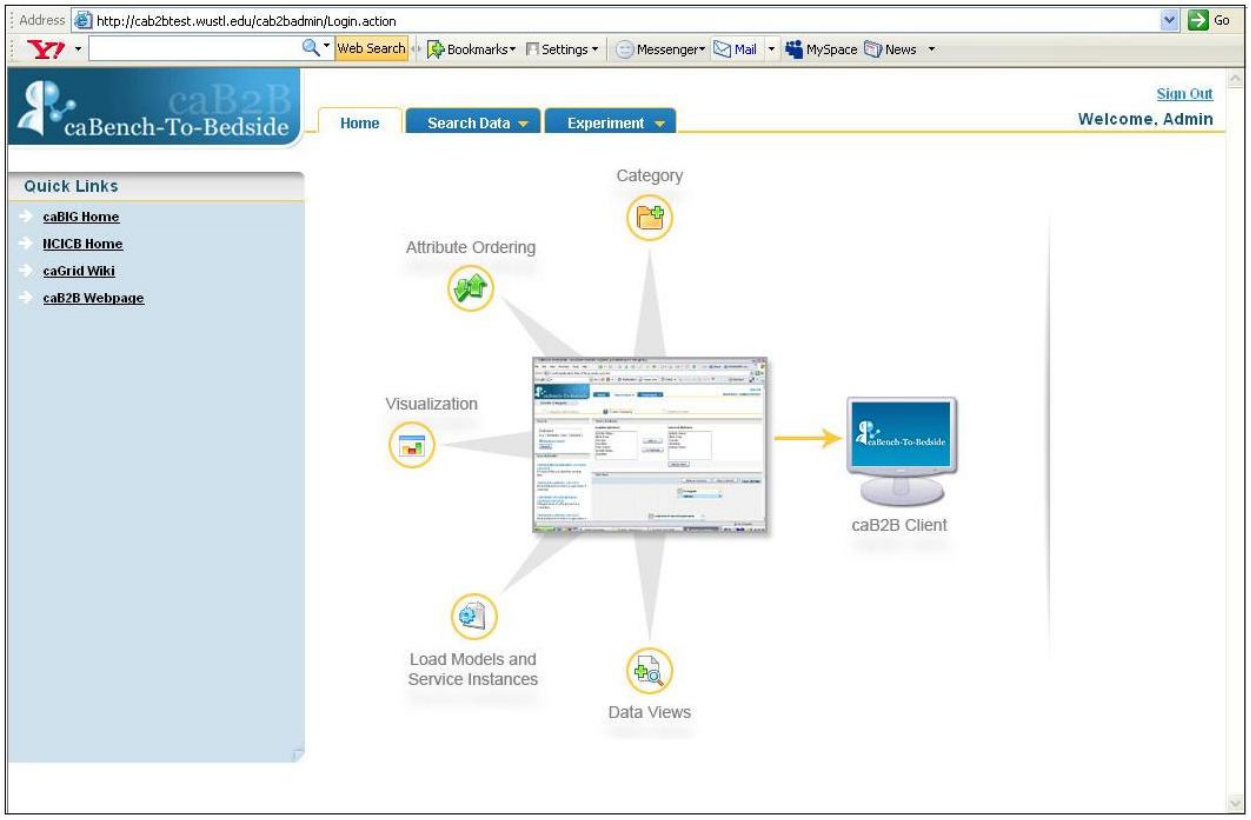

Figure 1.2. caB2BAdmin Module Home Page

The following chapters describe how an Administrator can use the Admin module to configure the caB2B client.

Document Text Conventions

The following table illustrates documentation conventions represented in this manual.

Convention	Description	Example
F.Y.I.	Indicates information of particular interest	F.Y.I. Administrator can create a new category.
CAUTION	Indicates that you should be particularly aware of the information provided.	CAUTION Categories once created cannot be modified

Chapter 2 Load Models

This chapter describes details of how to load models using the caB2B Admin Module.

This chapter contains the following topics:

- Introduction
- Loading Models

Introduction

The Load Models functionality is used to load models from the **ca**ncer **D**ata **S**tandards **R**epository (caDSR) into caB2B **M**eta**d**ata **R**epository (MDR). The caDSR is the central repository where all the models for the various caGrid-enabled services are stored.

The Load Models functionality fetches all the models available in the caDSR and lets you select and load a model into the caB2B MDR. Once you load a model into the caB2B MDR, the end-users can query the model through the caB2B client Interface. For example, loading the caArray model will enable end user to use caB2B client to query on caArray classes. These classes can be sample, experiment etc with attributes like experiment title and sample name.

What is a Model?

A model here refers to a Domain Model (also known as a UML Model). A domain model can be considered as a conceptual model of a system that describes the various entities involved in that system and the relationships between them.

The domain model is created to document the key concepts and the vocabulary of the system. The model identifies the relationships among all major entities within the system and usually identifies their important methods and attributes. Thus a model consists of the metadata which can be queried to retrieve the information stored in that model. For example, the domain model for caTissue service will have metadata regarding the categories like participant with attributes like first name, last name, gender etc.

What is Metadata Repository?

The Metadata Repository (MDR) stores the metadata for a model including its semantic annotations, such as all Common Data Elements (CDEs) including permissible values by decomposing the annotated UML model obtained from caDSR.

The MDR also contains all possible paths between each pair of entities. The caB2B server pre-calculates the paths between all pairs of entities in the UML model and stores them in the MDR. Categories from different applications are connected based on their attribute's CDE match. This involves matching the concept codes of the classes and their attributes in order. Finally, given the amount of information it

stores, it is also possible to get all the paths between two classes across two different UML models based on semantic interoperability.

Loading Models

To load models, log on into the Admin module and then follow the steps given below:

1. On the **Search Data** menu, click **Load Models**.

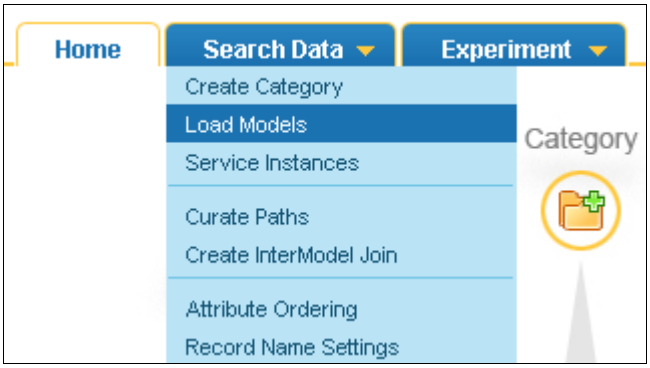

Figure 2.1. Search Data menu

2. The Load Model page opens, showing all available models in the caDSR, which have not been loaded in the caB2B MDR yet.

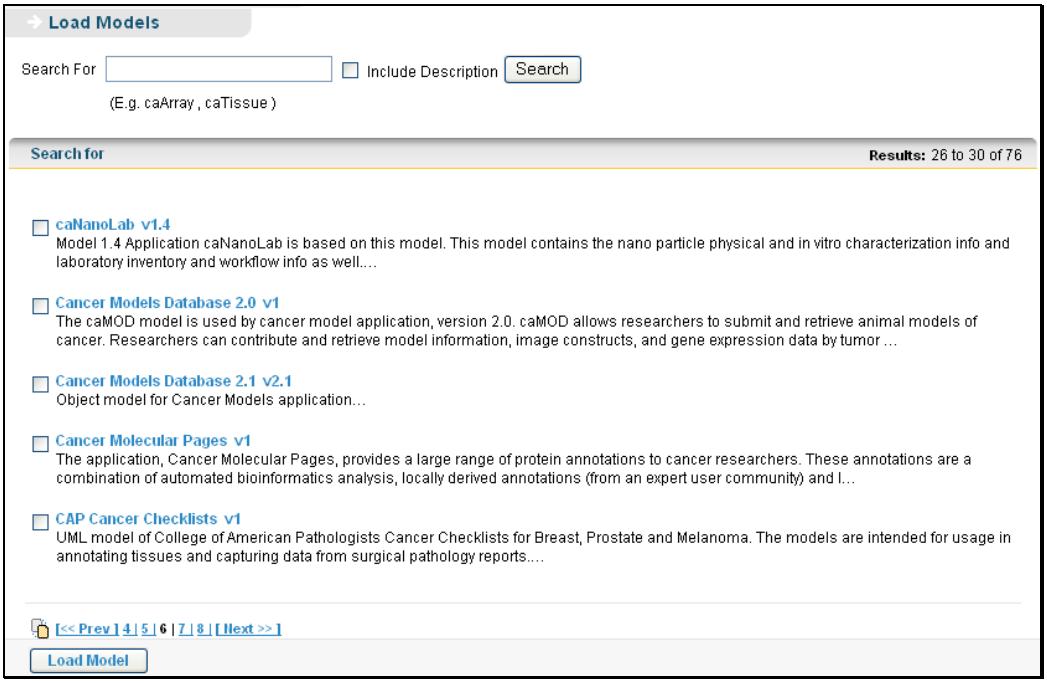

Figure 2.2. Load Models Page

F.Y.I.	Models once loaded successfully in the MDR are not visible in the available models list on the Load Models page. The same models with different versions are displayed,

3. In the **Search Box**, enter a search string to search for desired model and click **Search**. Select the **Include Description** check box if the search string is present in the model description.

Figure 2.3. Search for Models

4. In the search result, select the desired model and click **Load Model** to start loading the model.

F.Y.I.	To navigate back to view all matching models in the search result, click **Show All Models**.

Figure 2.4. Click Load Model to load a model

F.Y.I.	Once a model is loaded it cannot be removed or deleted.

5. While the models are loaded, a progress information page is displayed.

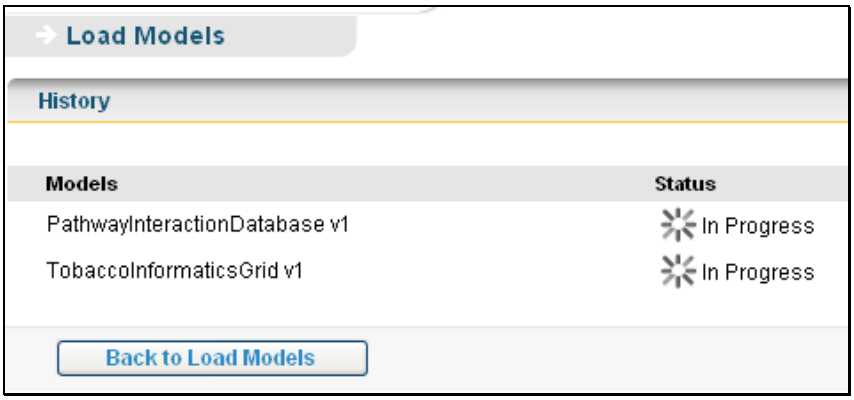

Figure 2.5. Model loading in progress

6. The **Load Model History** page is loaded when model starts loading into the system.

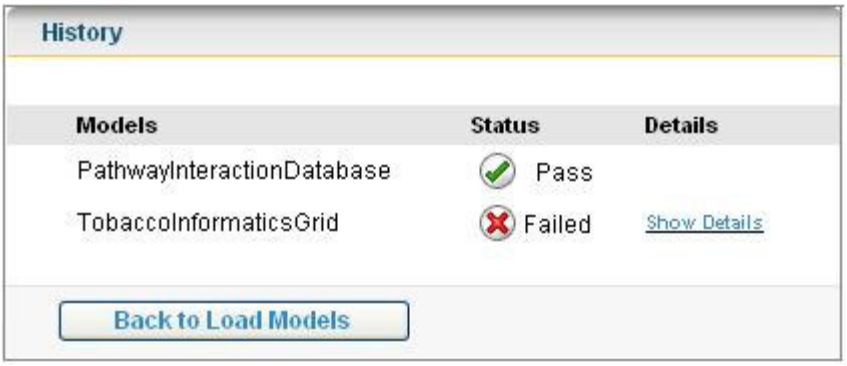

Figure 2.6. Load Model History

F.Y.I.	If a model fails while loading, then the reason for failure can be viewed by clicking the **Show Details** link. The details could

While loading models it is possible to select all the models at once and load them all together. Model loading generally fails when the model has not been properly annotated or there are some missing links in the model design and all the paths cannot be generated while storing the model in the MDR.

Chapter 3 Service Instances

This chapter describes the details of how to use the caB2B Admin module to configure the service instances.

This chapter contains the following topics:

- Introduction to the Service Instances
- Configuring the Service Instances

Introduction to the Service Instances

Service instances are active caGrid data services of any domain model (E.g. caTissue , caArray). A service instance has the metadata same as the corresponding domain model strored in caB2B MDR. You can consider a domain model as a template or guideline for the data and the corresponding service instance(s) as container(s) of the data. When an end-user queries a domain model, the corresponding service instance(s) gets queried and the data is fetched from the service instance(s).

The Service Instances functionality finds all the service instances executing at given point of time, which confirms to preloaded domain models. You can select multiple service instances to query for a model. For example, administrator can configure the application to save the *Washu caTissue instance* and the *UPitt caTissue instance* to be queried from the caB2B client. When an end-user queries a model from the caB2B client, the data would be fetched from all the instances specified/configured by the administrator using Service Instances functionality, this is one of the main rationale of caBIG.

The previous versions of the caB2B application used a property file to specify the service instances to query, but the property file has been deprecated in the current version 2.0 The instances to be queried are defined from the Admin module of caB2B and these instances are stored in the database.

F.Y.I.	It is important to define service instances, because if not configured the data cannot be fetched, hence the query cannot be processed.

Configuring the Service Instances

To configure the service instance to be used, first you must log into the admin module and load at least one model.

After you have loaded model(s), follow the steps given below:

1. On the **Search Data** menu, click **Service Instances**.

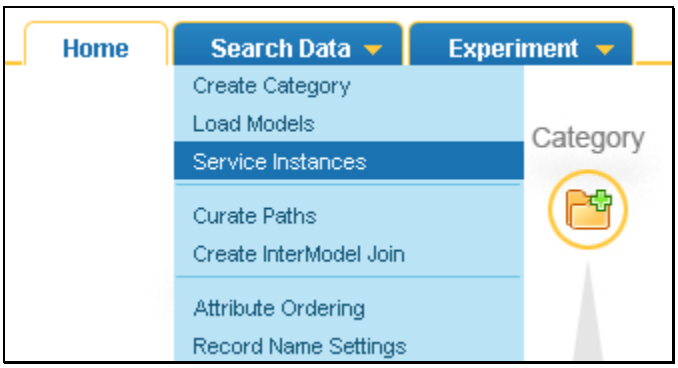

Figure 3.1. Search Data menu

2. The Service Instances page appears, showing all loaded models available in the MDR.

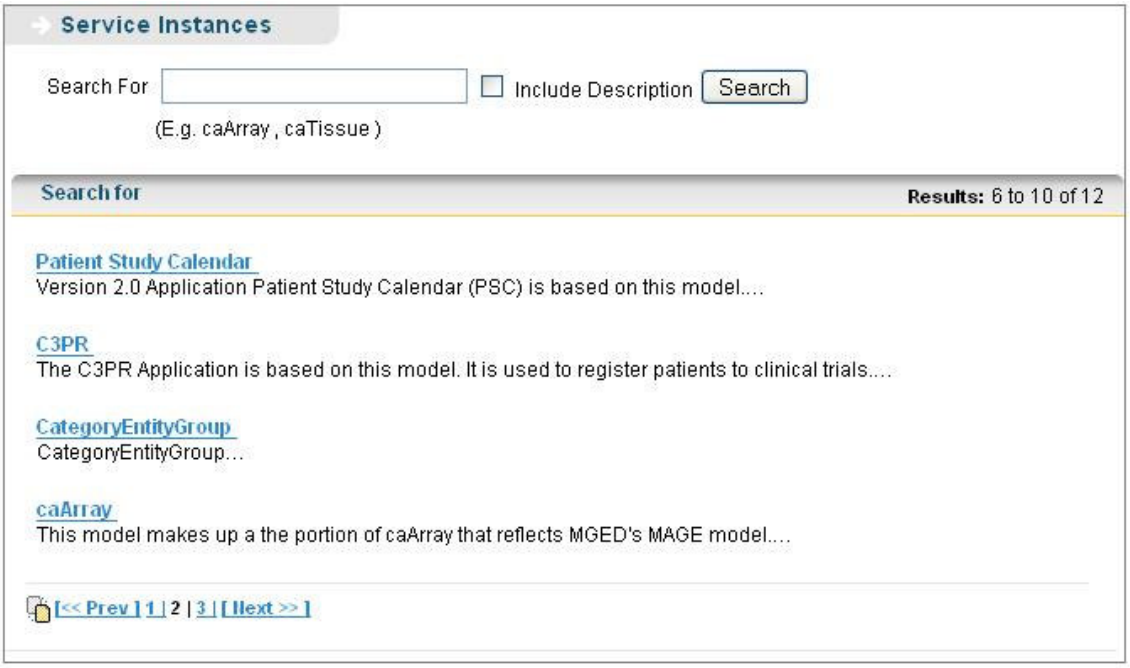

Figure 3.2. Service Instances Page

3. In the **Search For** box, enter a search string to search for desired model and click **Search**. Select the **Include Description** check box, if the search string is present in the model description.

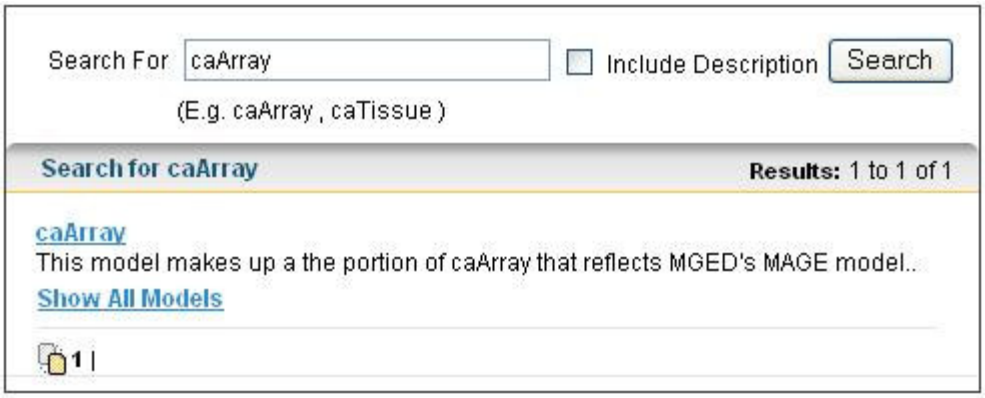

Figure 3.3. Searching a model

4. In the search result, click a model name to see all available service instances of that model. Service instance which are already configured are shown as checked.

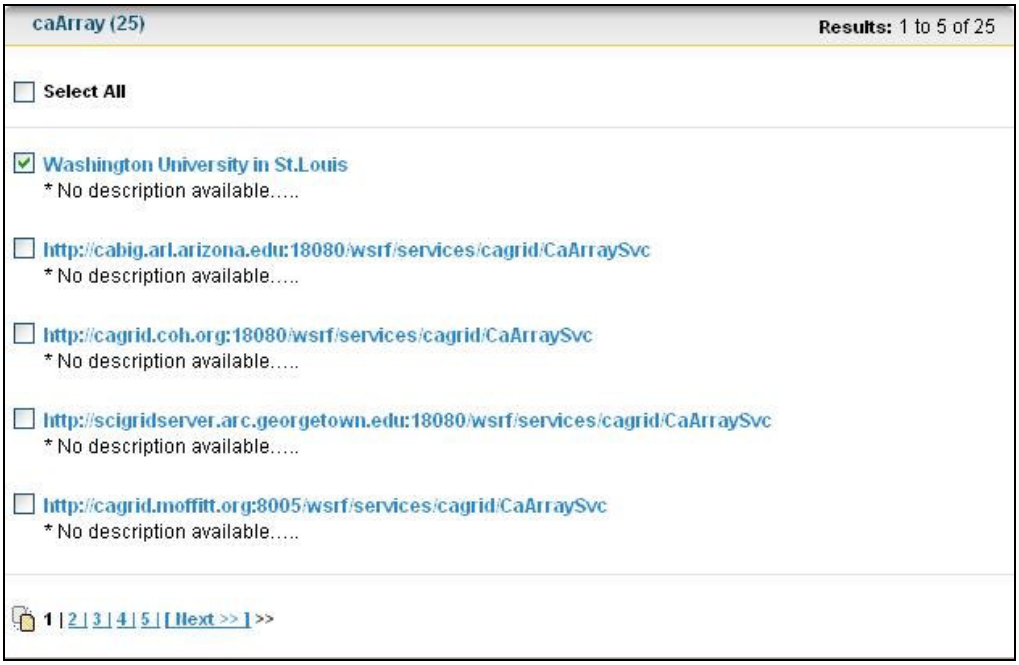

Figure 3.4. Service Instances currently running for the model

F.Y.I.	Click **Back** to navigate back to the Service Instances page without saving any of the instances

5. Select the corresponding check box to select a service instance and click **Save Settings**.
6. If the instance is saved successfully, a success message window is displayed. Click **Yes** in the message window to save another service

instance, or click **No** to exit.

Figure 3.5. Success Message

7. If the instance is not saved successfully, a failure message window is displayed.

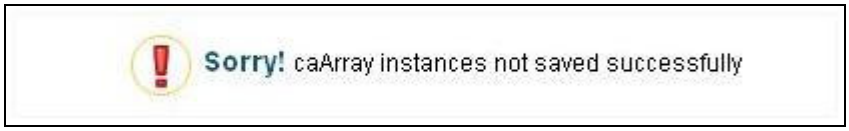

Figure 3.6. Failure message

8. User can remove an instance by deselecting the instance and clicking on **Save Settings**.

Chapter 4 Create Category

This chapter describes the details of how to use the caB2B Admin Module to create categories.

This chapter contains the following topics:

- What is a Category?
- What is a Path?
- How to create a Category?

What is a Category?

A category is a collection of *searchable* attributes. You can use a category to search, view, and add data to experiments. A category can be a combination of several entities or domain objects. Also, every domain object in an application is basically a category.

For example:

1. Participant, a domain object in the caTissue Core application, is a category consisting of the following attributes:
   - Identifier
   - First Name
   - Last Name
   - Gender
   - Race
2. Gene Annotation, a domain object in the GeneConnect application, is a category consisting of the following attributes:
   - Entrez Gene ID
   - Gene Symbol
   - Gene Name
   - GenBank Accession Number
   - Taxonomy ID
   - Chromosome

The categories Participant and Gene Annotation in the examples above refer to a single entity respectively. You can use the Admin module to define categories which refer to more than one entity. You can combine two or more entities, like participant and specimen, to create a new category. This new category can be customized to have only those attributes from each entity that are generally used in queries.

For example, an end-user wants to query for all the participants whose specimens are available. Since specimen availability is not an attribute of the Participant class, you can create a category, in which the specimen availability attribute from the

Specimen entity is combined with the attributes from the Participant class. Now the end-user can query a single category with all the desired attributes.

This way, you can have many categories with all the desired attributes from different entities combined together to form a single category.

What is a Path?

A path tells the system how two categories are to be connected. A path is one of the multiple ways in which one category can be connected to another. Since there can be multiple paths that connect two categories, the results of a query will differ based on the path you select. Therefore it is important to choose the right path for your query.

For example, consider the two categories Specimen and Site (from the caTissue Core application). A Specimen is associated with the Site category in two ways:

1. Site where the specimen was collected
2. Site where the specimen is stored

That is, there are two paths between the categories Specimen and Site: 1) Through Specimen Collection Group for the collection site, 2) Through Storage Container

Query example 1: Show all the available specimens collected in Barnes Jewish Hospital. Since you are specifically interested in a particular collection site in this query, you must choose the path through Specimen Collection Group.

Query example 2: Show all the available specimens stored in Barnes Jewish Hospital. Since you are specifically interested in a particular storage site in this query, you must choose the path through Storage Container.

What is an Ambiguity Resolver?

There are certain categories between which multiple paths exist. These paths provide the end-user with the option of connectivity between the categories via different routes. It is possible that depending on the situation the end-user may wish to choose different paths. In such a situation where multiple paths exist, the end-user is shown an Ambiguity Resolver window when the end-user tries to connect the two categories though the **Manual Connect** option. Then the end-user can select one of the available paths from the Ambiguity Resolver window to connect the two categories.

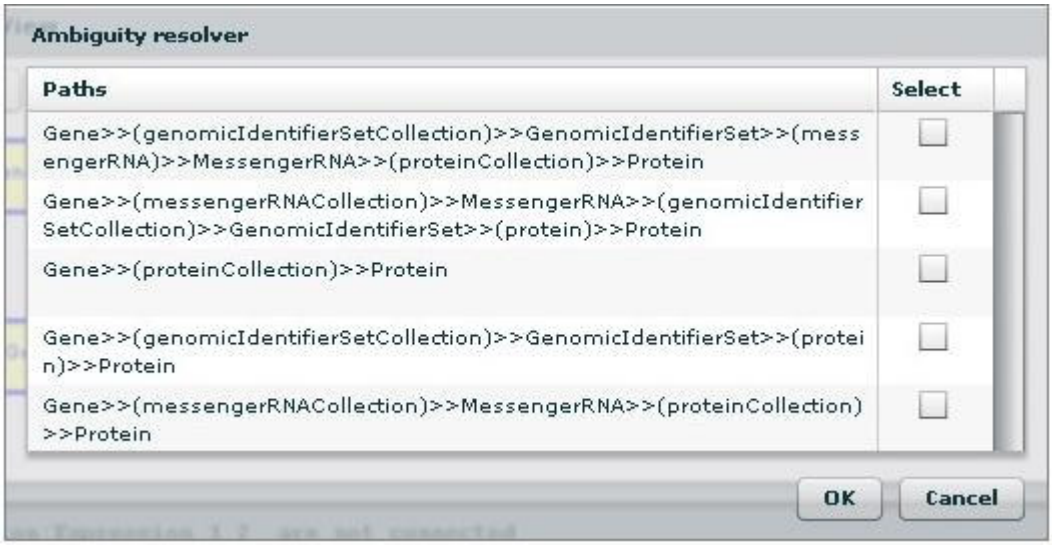

Figure 4.1. Ambiguity Resolver window

How to create a Category?

To create a category, start with logging into the Admin module and then follow the steps given below:

1. On the **Search Data** menu, click **Create Category**.

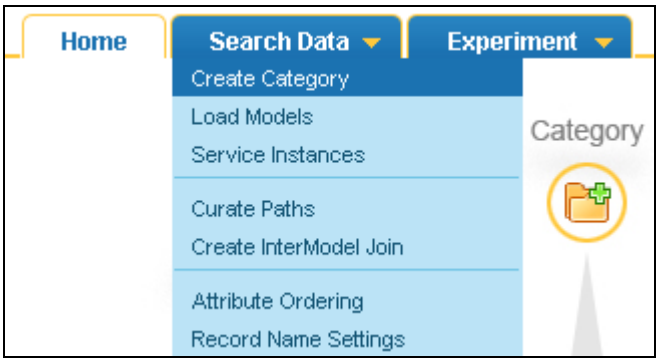

Figure 4.2. Search Data menu

2. The Create Category page appears, showing three tabs: **Category Information**, **Create Category**, and **Attribute Order**.

Figure 4.3. Category Creation process

3. In the **Category Information** tab, enter a category title in the **Category Title** box. Enter the category description in the **Category Description** box.

Figure 4.4. Category Information tab

4. In the **Create Category** tab, perform the following steps:
   a. Enter a search string in the search box and click **Search**.

Figure 4.5.Searching for metadata

   b. In the **Search Results** area, click the class name to be added to the category. The attributes of the selected class are seen in Available Attributes list.
   c. In the **Available Attributes** list, select the attributes to be present in the category (To select multiple attributes, hold down the CTRL key while selecting the attributes).

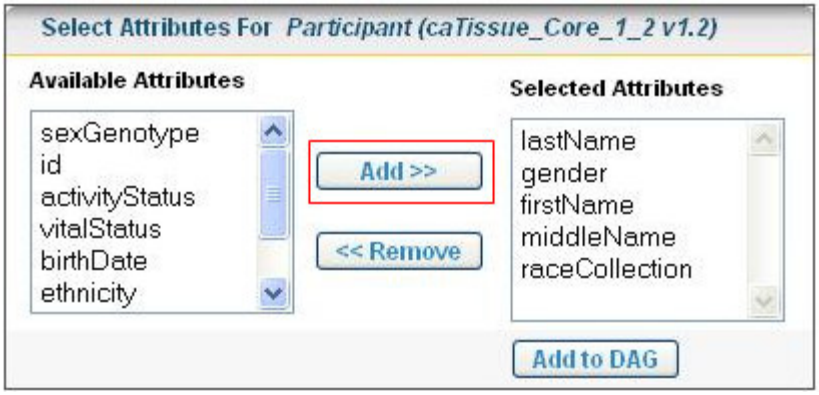

Figure 4.6. The Entity Panel

d. Click **Add** to move the attributes to **Selected Attributes** list.

	In the **Selected Attributes** list, select the attributes to remove from the category. Click **Remove** to remove the attributes from the Selected Attributes list.

e. Click **Add to DAG** to add the category to **DiAGrammatic** View.

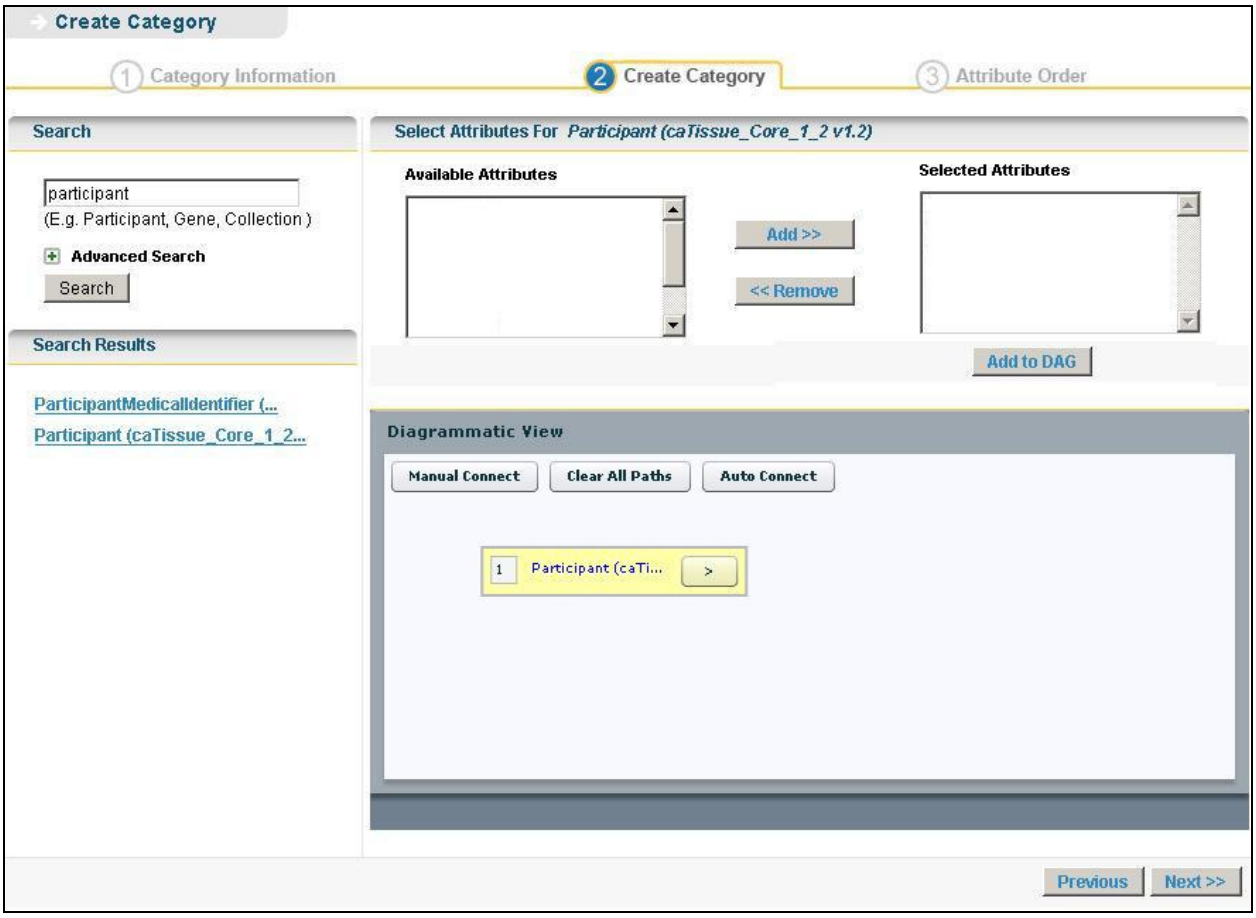

Figure 4.7. The DAG Panel

	The **DiAG**rammatic View (DAG) panel is similar to a canvas where all the classes appear in the form of nodes. The system adds a yellow rectangle or node that represents the class being added to the DAG view. As you continue to add classes, the system adds corresponding rectangles to the DAG view.

f. Repeat the steps *a* to e to add all required categories.

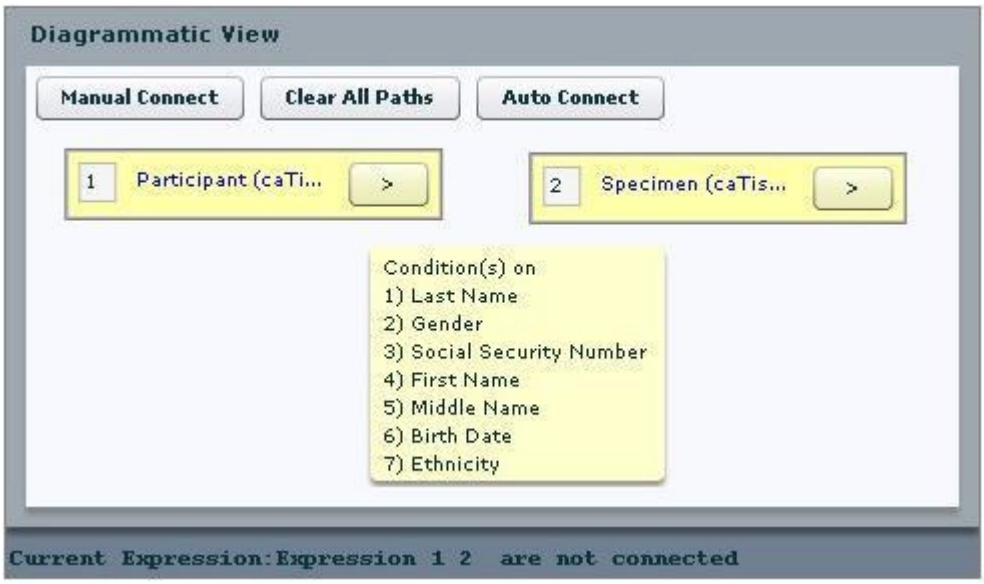

Figure 4.8. Diagrammatic View

F.Y.I.	Clicking the arrow > gives options to edit the conditions or delete the class added to DAG.

g. Click **Manual Connect** to connect the categories. If multiple paths exist between the categories, then Ambiguity Resolver window is displayed. In the Ambiguity Resolver window (see Figure 4.1), select the path.

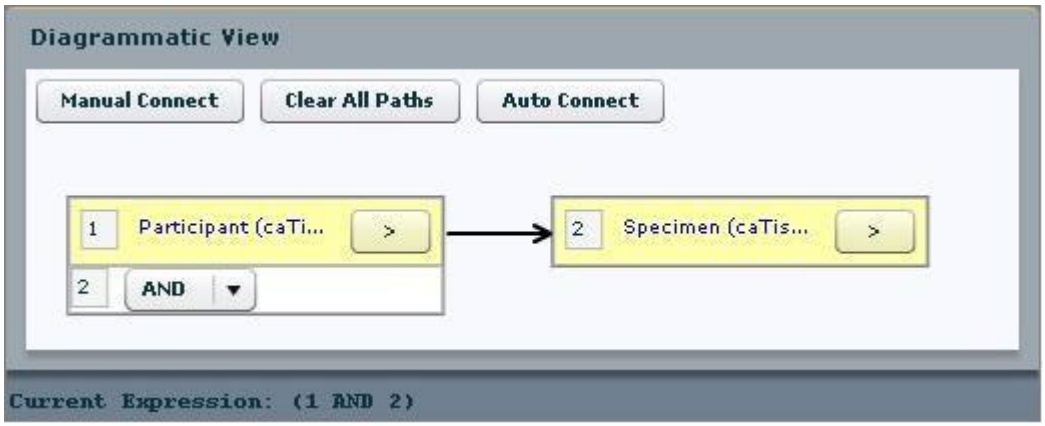

Figure 4.9. DAG with nodes connected

F.Y.I.	Click **Auto Connect** to connect the classes when there exists curated paths(see Use of the Curated Path)

h. Once all classes are connected, click **Next** to navigate to the **Attribute Order** tab.

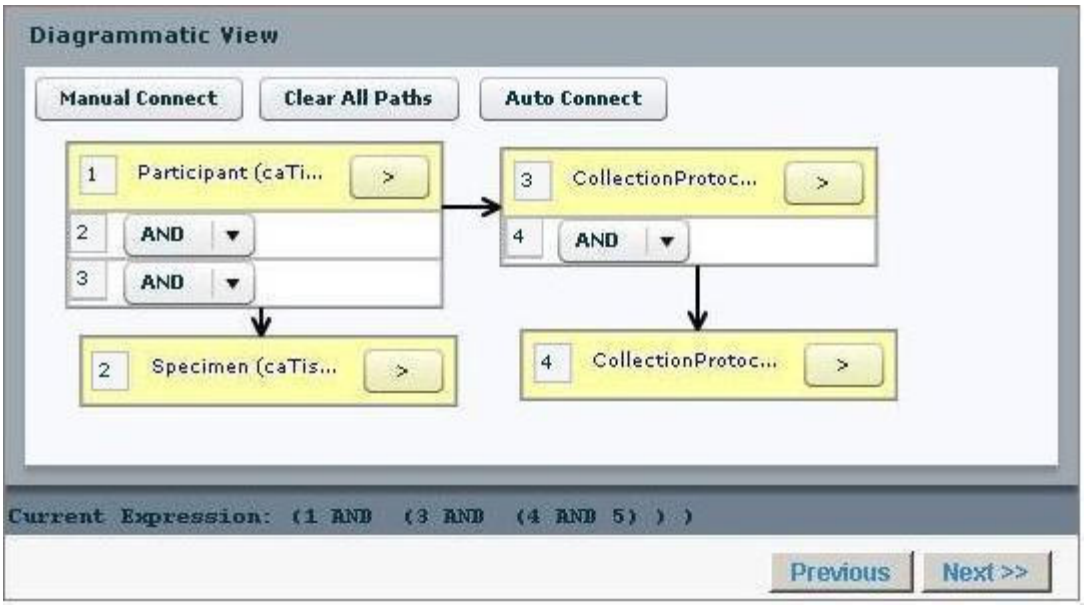

Figure 4.10. Four nodes connected in DAG

5. The ordered list of attributes is displayed on **Attribute Order** tab, where display label can be changed if required. Click **Finish** to save the category.

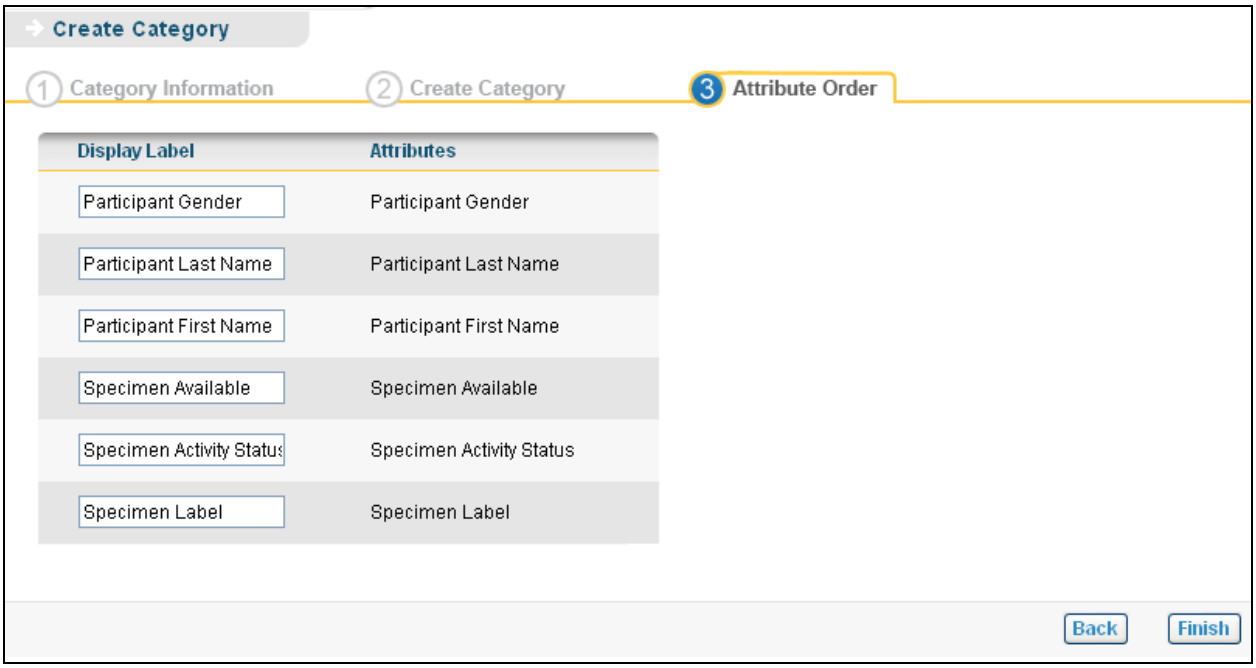

Figure 4.11. Attribute Order tab

6. If the category is created successfully, a success message window is displayed. Click **Yes** in the message window to create a new category, or click **No** to exit.

Figure 4.12. Success Message

7. An error message is displayed if the category is not saved successfully.

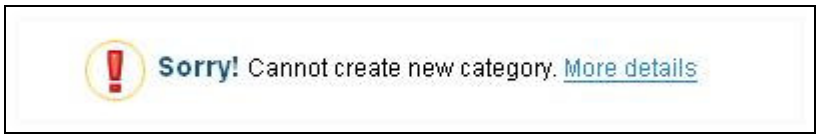

Figure 4.13. Failure Message

Advanced Search

Expanding the **Advanced Search** box displays the advanced options that help you refine the category search. The following is the default selection of the various options:

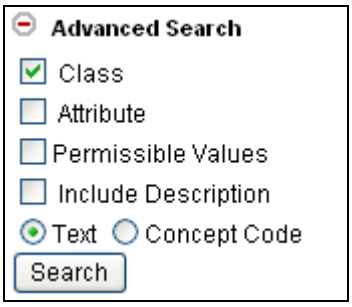

Figure 4.14. Advanced search

If the system returns too many results for your class search, you can use the **Advanced Search** to narrow your search. The Advance Search options are explained below:

Option	*Description*
Check Boxes	
Class	Includes the category names in the search.
Attribute	Attribute (also known as data element) is the lowest unit of the searchable items. A class contains many attributes. Select this check box to include the individual attribute names in the

Option	Description
	search. For example: If you select the *attribute* check box and enter search string *gender*, the search result will include the category Participant since it contains an attribute by name *Gender*.
Permissible Values	Some attributes contain a fixed set of values. For example, Gender can be Male, Female or Unspecified. Organ can be heart, lung, breast etc. This fixed set is known as **Permissible Values**. Select this check box to include the permissible values in the search. For example: If you select the Permissible Values check box and enter search string *Male,* the search result will include the Participant category. The Participant category contains an attribute *Gender* which contains a permissible value *Male gender*.
Include Description	Includes the description text in the search. **Note: Include Description** check-box will be enabled only for a text-based search when you select the **Category** check box**,** the **Attribute** check box, or both.
Options	
Text	The system matches the text that you type as the search string with the name of the category, its attributes, or the permissible values. This is the default search option and a simple, most commonly used way of searching a category.
Concept Code	The system treats the search string that you type as an Enterprise Vocabulary Services (EVS) concept code. The system matches this concept code with the concept code of the category, its attributes, or the permissible values. The results contain matching categories. More details in concept code can be found on http://nciterms.nci.nih.gov/NCIBrowser/SearchConcept.do and http://ncimeta.nci.nih.gov/MetaServlet/

Chapter 5 Curate Paths

This chapter describes the details of how to use the caB2B Admin Module to curate paths.

This chapter contains the following topics:

- Introduction to the Curated Paths
- Creating a Curated Path
- Using Curated Path

Introduction to the Curated Paths

You already know that a path is one of the multiple ways in which one class can be connected to another class. There can be multiple ways to connect two categories and it is possible that only a few of these multiple paths available are useful while querying the domain model. As the administrator, you must be able to define the most useful and frequently used paths to be selected by default, which are known as curated paths. These could be the paths which are selected most of the times while querying. The curated paths reduce the caB2B users' task of selecting the most suitable and logical path from the available paths.

For example, multiple paths are available to connect the two classes Specimen Collection Group (caTissuecore v1.2.2) and Specimen (caTissuecore v1.2.2). Most of these paths consist of one of the event parameters classes in the path, like fixed event parameter or transfer event parameter. A user may not be always interested to know if the specimen has been thawed or if the specimen has been fixed or it has been embedded. Most of the times what the user is really interested to know is what kinds of specimens are available or those that have been collected and are available for use. For finding such specimens the user can just query on the Specimen class and connect it directly to the Specimen Collection Group.

As the administrator you can select one of the available paths which would connect the two or more classes and exclude the rest of the paths. This way, the paths between the classes are curated. When connecting the classes, the end-user is always shown the curated paths first, and this saves the user from the hassles of selecting a correct path from the available list of multiple paths. The curated path is always given a preference over all the other available paths, and is selected by default when connecting categories. If only a single curated path exists between categories, the classes are always connected via this curated path.

Creating Curated Paths

You can define a curated path between two or more classes. All the classes used to define a curated path should be from the same domain model.

For example, it is possible to define a path between Participant, Participant Medical Identifier, Collection Protocol Registration, and Collection Protocol categories; since they are from the model caTissuecore v1.2.2. But a curated path cannot be defined between the classes from different models like Gene from caFEv1.1 and Gene from GeneConnect as these are from the different models.

To curate a path, start with logging into the Admin module and then follow the steps given below:

1. On the **Search Data** menu, click **Curate Paths**.

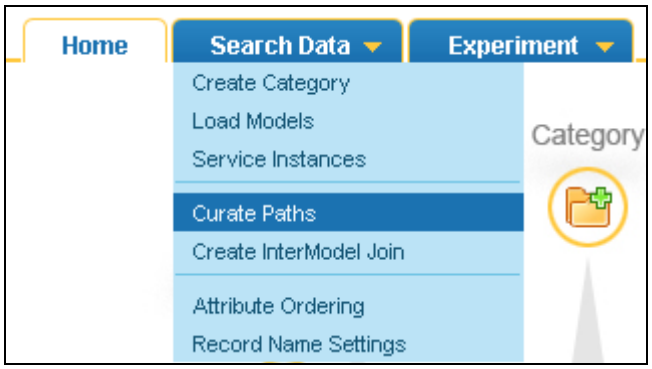

Figure 5.1. Search Data menu

2. Enter the search string in the search box.

Figure 5.2. Text box to enter the search string

3. In the **Search Results** area, click the category name to add it to the Diagrammatic View. Add two or more categories between which the path is to be curated.

Figure 5.3. Select Class to add in DAG

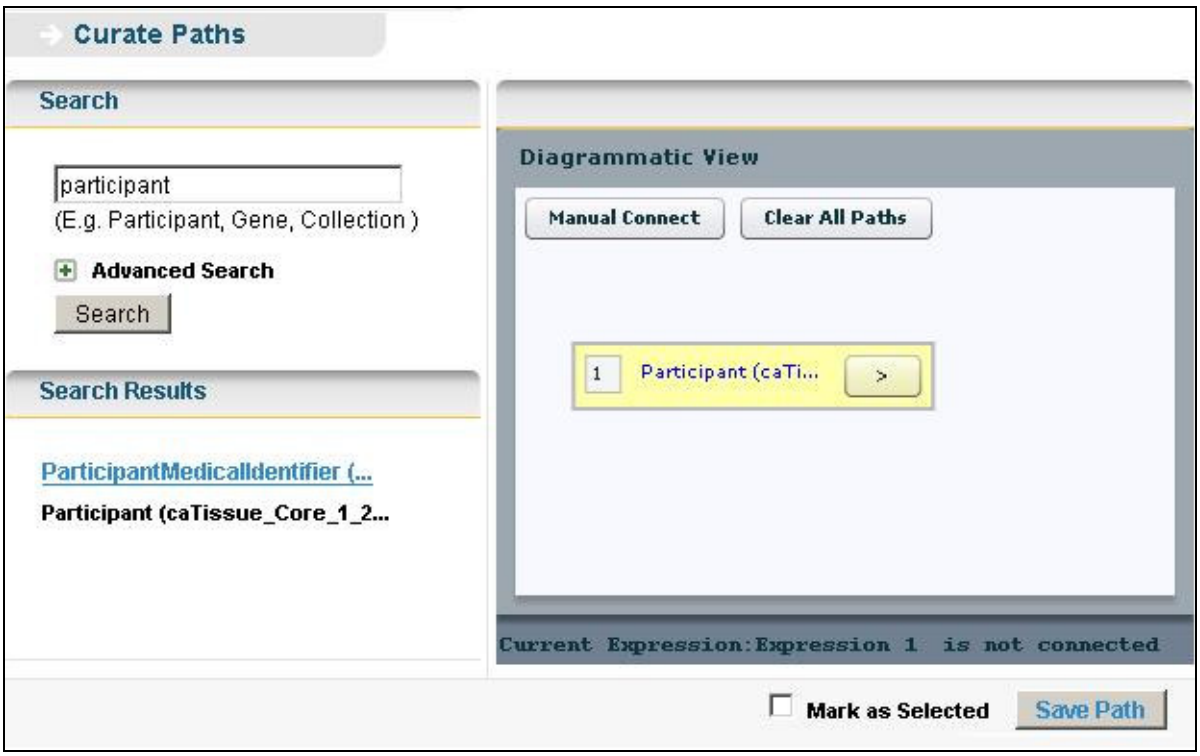

Figure 5.4. The DAG Panel

4. In the Diagrammatic View, select a pair of classes and click **Manual Connect** to define the path between them.

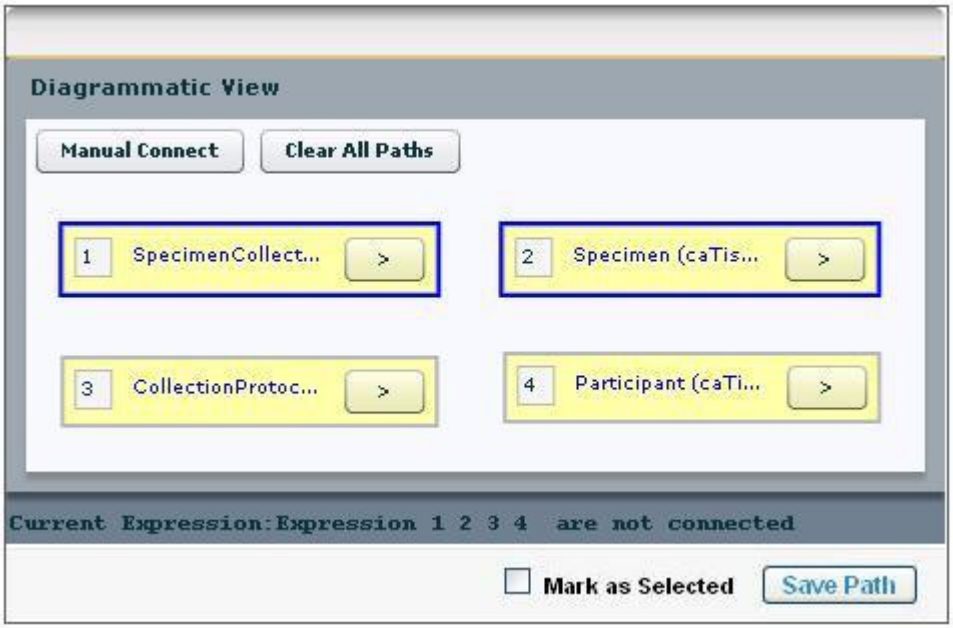

Figure 5.5. Defining path between a pair of classes

5. Similarly connect all the categories listed in the Diagrammatic View.

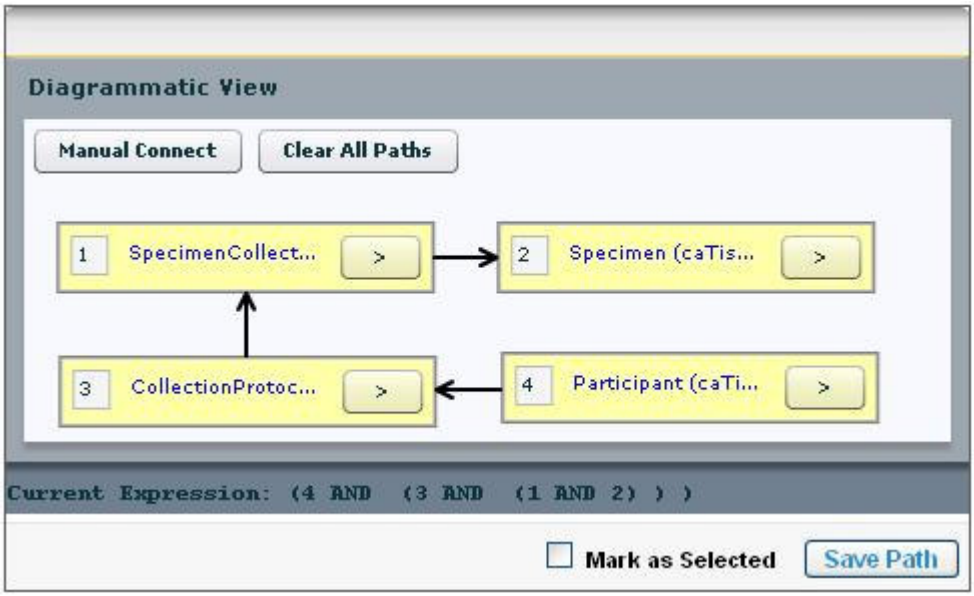

Figure 5.6. All classes connected to form a complete graph

6. Select **Mark as Selected** check box, if you want this path to be chosen by default while querying. Once a path is marked as selected no other paths are shown to the user.
7. Click **Save Path** to curate and save the path present between the categories.

8. If the curated path is stored successfully, a success message dialog box is displayed. Click **OK** in the dialog box.

Figure 5.7. Success Message

9. If the path already exists, a warning message dialog box is displayed and the path is not saved again.

Figure 5.8. Failure Message

Use of the Curated Path

A curated path can be used while querying as well as while defining a category. In both the cases the same method is used to connect the classes. The steps used are outlined below:

1. Using curated path to define category:
   a. To create categories, refer to the topic “How to create a Category?” upto step 4.g, in the **Chapter 3 Create Categories**.
   b. Select all the classes between which a path is defined. Then perform step i if there are more than 2 classes in DAG else perform step ii or iii.

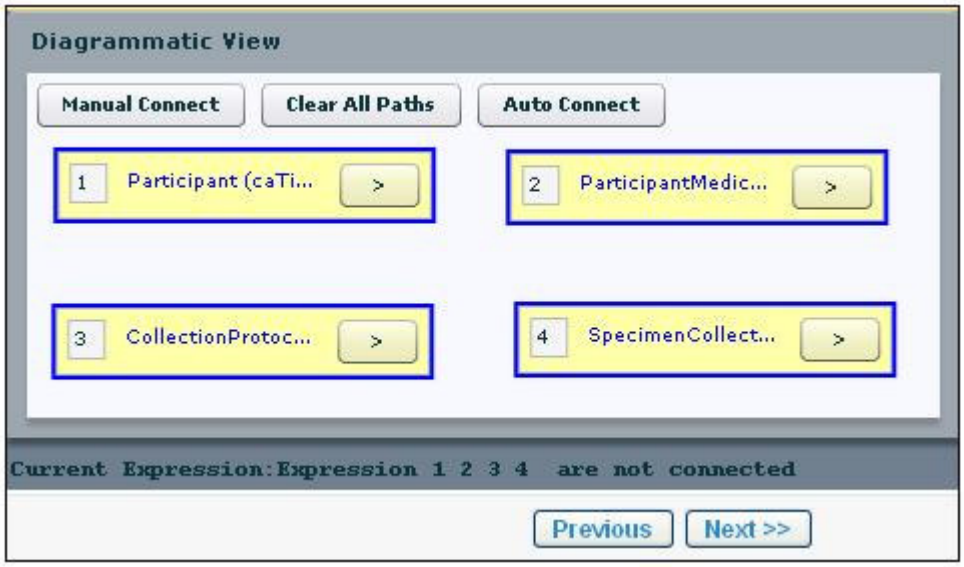

Figure 5.9. Select all nodes

i. Click **Auto Connect** to connect all the classes with the defined curated path. If multiple curated paths exist, then the Ambiguity Resolver window is displayed to select a single curated path.

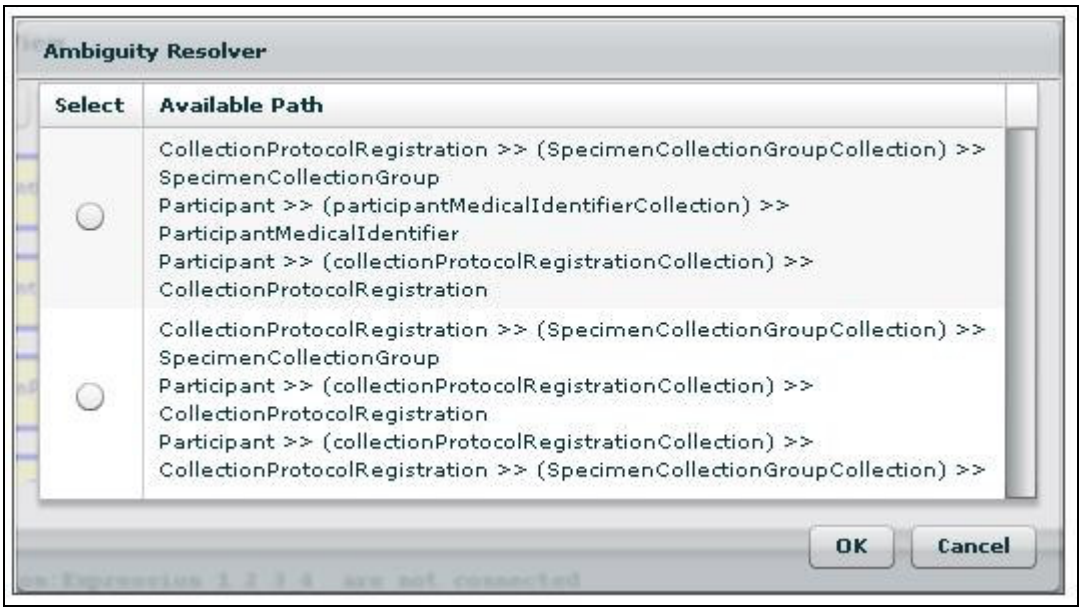

Figure 5.10. Ambiguity Resolver window for multiple curated paths

ii. Click **Manual Connect**, an Ambiguity Resolver window is shown with the curated paths as the default view, select the path and save the category.

F.Y.I.	Manual connect works when there are only two classes present in the Diagrammatic View.

Figure 5.11. Curate path as default view

iii. Click **All Paths** in the Ambiguity Resolver window to select a path other than the curated path and save the category.

2. Using curated path while querying:
   a. While querying the applications in caB2B client, specify the limits(Refer to section Define Limits in caB2B v2.0 end user manual).
   b. Select all the classes in DAG and click **Auto Connect** to connect the classes with the curated path.

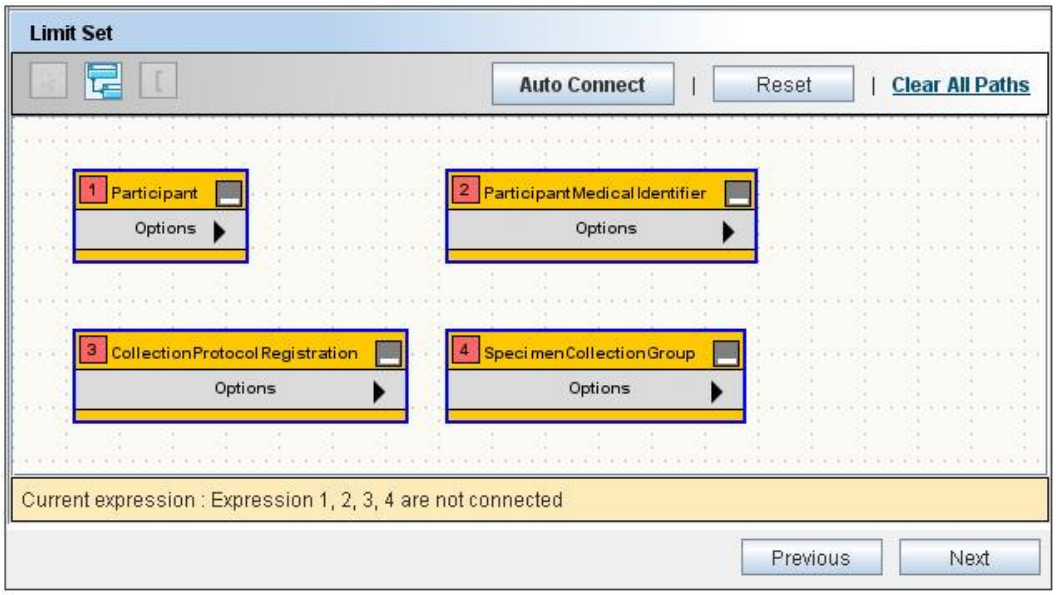

Figure 5.12. Selected nodes in caB2B client

c. If the path has been marked as selected then that path is used by default and the classes are connected.

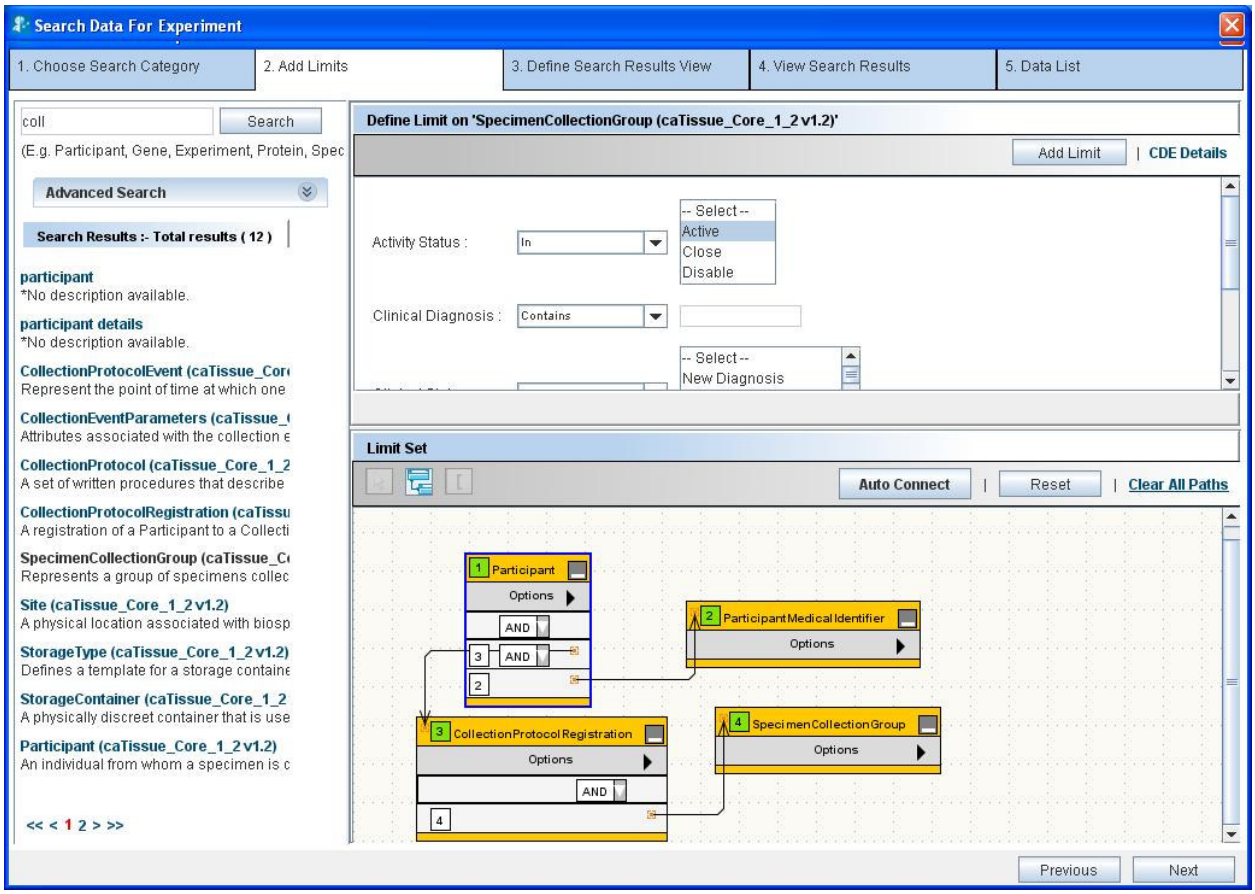

Figure 5.13.classes connected in DAG during query as per curated paths

d. If multiple paths exist then an ambiguity resolver window is shown, select the most appropriate path to connect the classes.

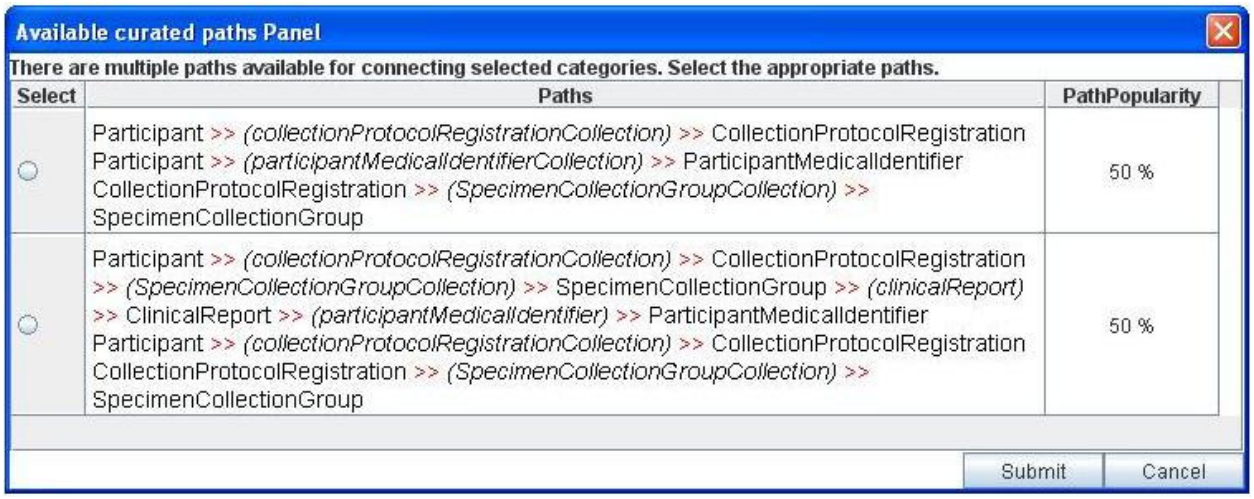

Figure 5.14. Ambiguity resolver for more than one curated paths

e. A message is displayed if no curated paths exists.

Thus, a curated path once defined can be used to connect the classes directly which helps the end-user save time.

Chapter 6 Inter Model Joins

This chapter describes the details of how to use the caB2B Admin Module to create inter model joins.

This chapter contains the following topics:

- Need for Inter model joins
- Introduction to Inter model joins
- Creating a inter model join
- Using inter model join

Need for Inter Model Joins

There are many domain models which share some common data. But due to the unavailability of interoperability between the models, it is not possible to get the data from across the models. To connect the different models and share data among them, it is necessary to have a common link between these different models.

For example, a translational scientist collects a sample and stores the sample information in a tissue banking software, such as caTissue. Later he performs a microarray experiment on the sample and stores this experiment data in the caArray repository. In future, when he/she wants to know the details of the specimen used in a microarray experiment, then he/she has to separately query the caTissue and caArray to retrieve the information. If there is a connection defined between caTissue and caArray model based specimen then he/she can directly retrieve the desired information with a single query. This way one can also use common attributes like Gene ID to query between tools like caFE and GeneConnect. To provide this kind of inter model compatibility, you need to define inter model joins.

Introduction to Inter Model Joins

Defining an Inter model join is similar to defining a curated path between classes from two different models. This join defines a connection between any two models.

For example, consider two models GeneConnect and caFE. Both of these models have a class called Gene having attributes Id, Symbol, Name, Summary, Pubmed count and chromosome map for caFE gene and Id, EnterzGeneId, EntrezGene as Output,etc. from GeneConnect.

If a user wants to retrieve all the genomic identifiers for a gene, for which the user knows the gene symbol, then he has to query all the databases individually and retrieve, gather and curate information from all the sources. This task can be simplified by defining inter model joins. In an inter model join, two categories are connected based on the common attributes, i.e., if the two classes have some common attributes with same concept codes, then joins are defined on such attributes.

While curating path across two models, the administrator must add the classes, which are the end points of inter-model bridge. The two classes are connected by the public ID of the attributes, which is the same as Data Element public ID as

present in the Common Data Element (CDE) browser (http://cdebrowser.nci.nih.gov/CDEBrowser/). The classes can also be connected by matching Data Element Concept (DEC), which includes matching the concept codes of the attributes and the classes as present in the NCI Metathesaurus (http://nciterms.nci.nih.gov/NCIBrowser/Dictionary.do). An administrator can also define the attributes manually which should be used to bridge the two classes.

Creating an Inter model Join

An inter model join can be defined only between two classes from different models, and it acts like a bridge connecting two separate models. To define an inter model connection, start with logging into the Admin module and then follow the steps given below:

1. On the **Search Data** menu, click **Create InterModel Join**.

Figure 6.1. Search Data menu

2. Enter the search string in the search box.

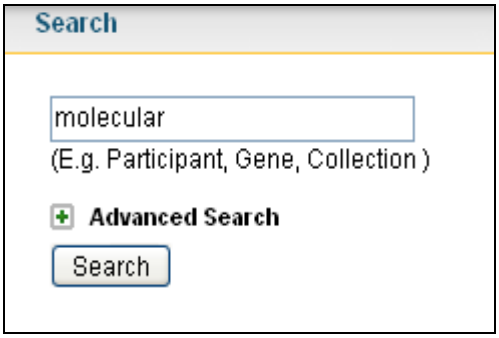

Figure 6.2. Search box

3. In the **Search Results** area click on the **class name**, which is to be added in the DAG.

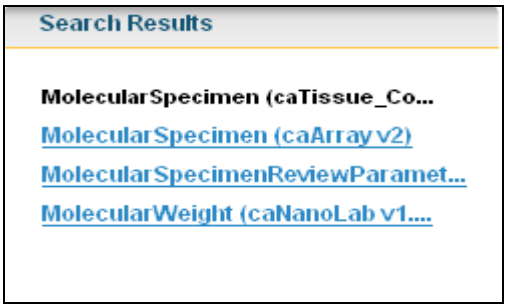

Figure 6.3. Search results

4. The class appears in the DAG. Add the second class to the DAG.

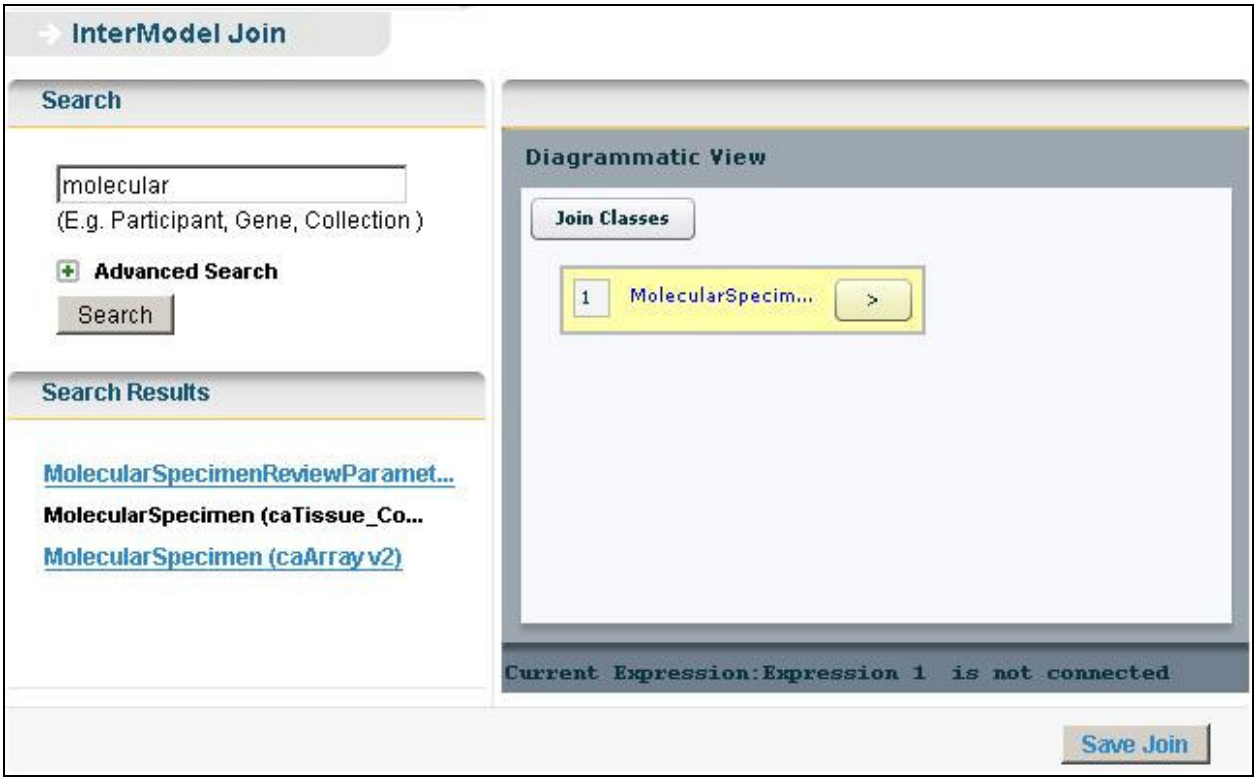

Figure 6.4. DAG Panel

5. Select both the classes and then click on **Join Classes**.
6. From the pop-up window select one of the many paths.

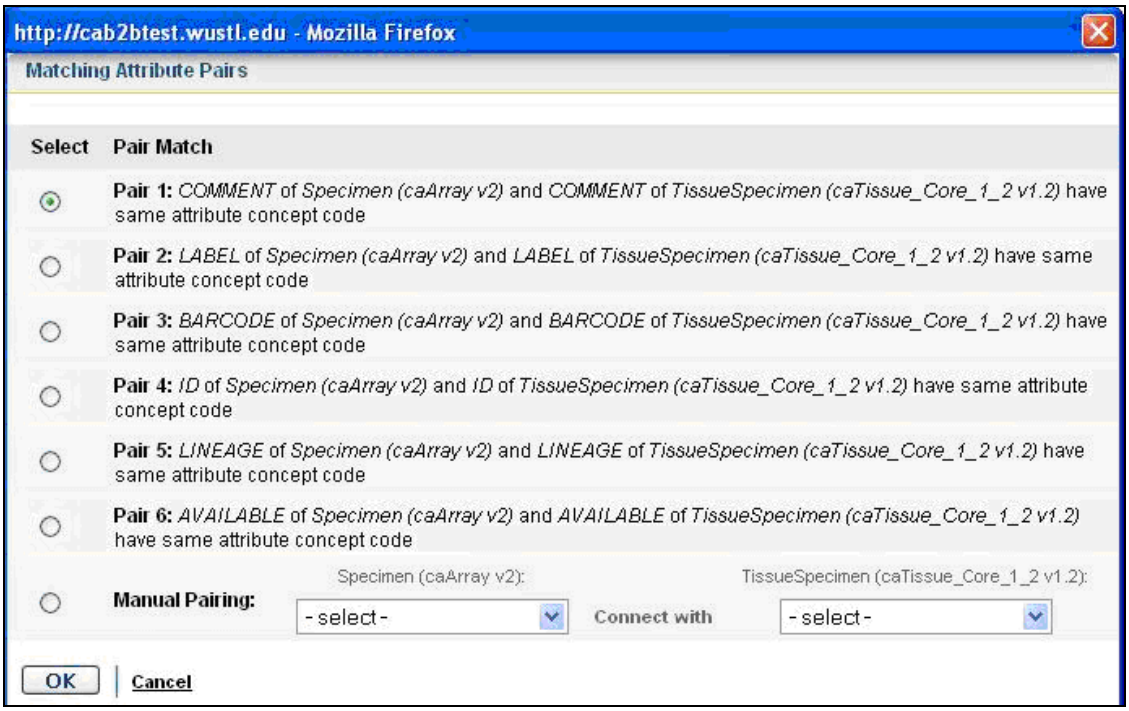

Figure 6.5. Pop-up window displaying the paths

7. If no paths are available or to override the available paths select the attributes to be connected manually, by selecting attributes from the option **Manual Pairing**.
8. Click **OK** to save the selected path.

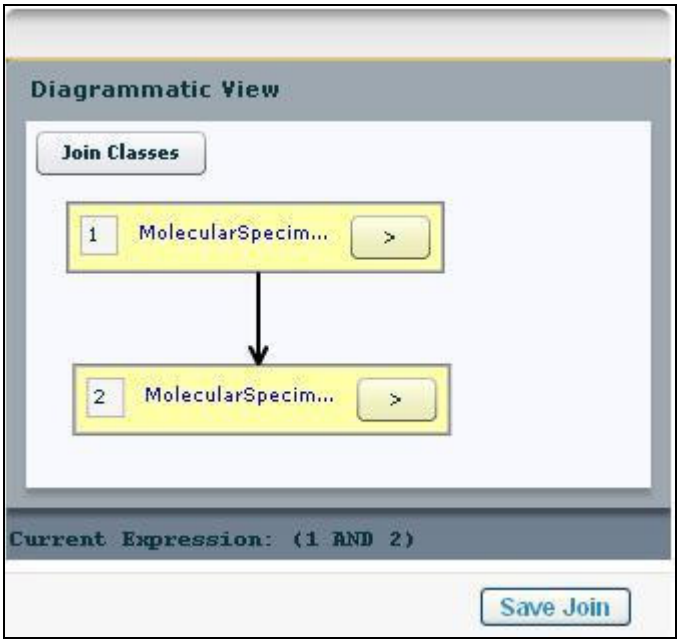

Figure 6.6. Classes connected in DAG

9. Once the nodes are connected, click **Save Path** to save the inter model connection.
10. If the inter model join is saved successfully, a success message is displayed.

Figure 6.7.Success Message

11. If a duplicate inter model join is defined, then an error message is displayed.

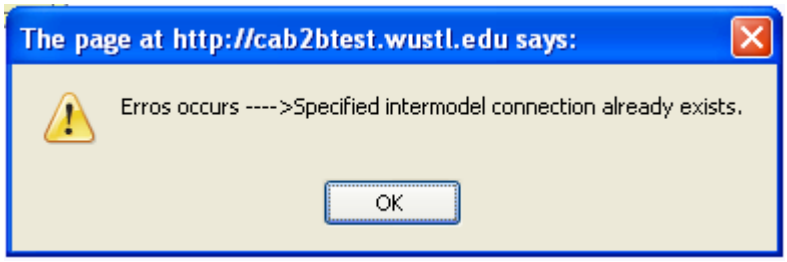

Figure 6.8.Failure Message

Using Inter model Join

These inter model connections once defined can be used while creating categories or while querying the caB2B client (refer to Use of the Curated Path). This path can be used to query across the models by connecting two classes from different models.

Example : if admin creates an inter model join between gene(caFe) and gene (Geneconnect),then this path can be used for querying. The two classes are joined through the defined path.

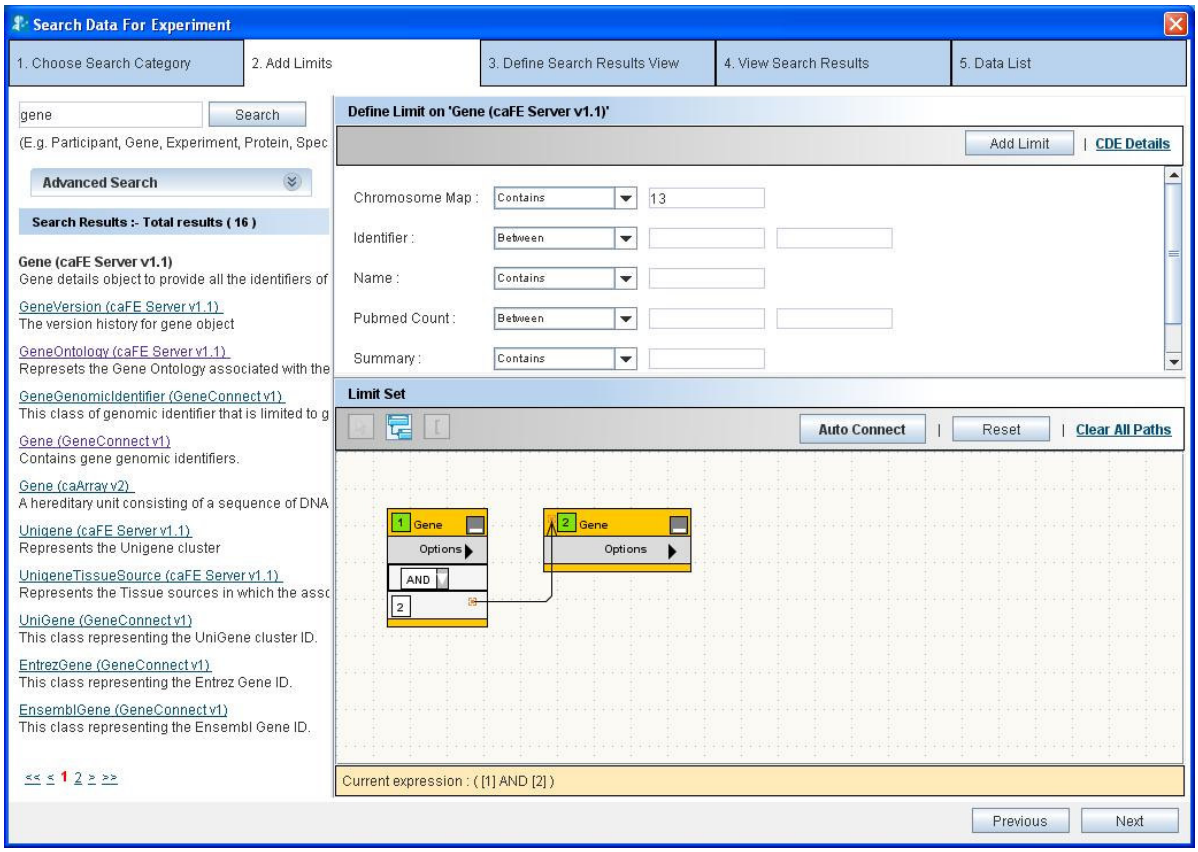

Figure 6.9. Classes connected in DAG as per the inter model joins defined

Index

admin manual 1
administrator 3
Advanced Search ..... *See* Search
Ambiguity Resolver 17, 18, 22, 31, 32
attributes .. 8, 16, 17, 19, 20, 23, 24, 25, 35, 38
Auto Connect 31, 32
caB2B 1, **3**, **4**, **5**, **8**, **9**, **12**, **16**, **26**, **32**, **35**, **39**
caDSR 3, 8, 9
caFE 35
caGrid 3
cancer Bench-To-Bedside *See* caB2B, *See* caB2B
cancer **D**ata **S**tandards **R**epository ... *See* caDSR
categories ..... *See* category
caTissuecore v1.2.2 26
classes 8, 21, 26, 28, 30, 31, 32, 33, 34, 35, 36, 37
Collection Protocol categories 26
Collection Protocol Registration 26
Common Data Element ..... *See* CDE
Common Data Elements ..... *See* CDE
configure the service instance 12
Creating Experiments 12
Curate 4, 5, 26, 27
Curate Paths 26
Data Element public ID 35
Di**AG**rammatic View ..... *See* DAG
Enterprise Vocabulary Services ..... *See* EVS
entities ..... *See* categories
EVS 25
Experiments
  creating 12
GeneConnect 35
Home Page 6
Installation Guide 6
instances ..... *See* service instances
inter model join 35
Join Classes 37
joins ..... *See* Inter Model Joins
Login 3, 5
Login Page 5
Manual Connect 17, 22, 28, 31
Manual Pairing 38
MDR 8, 9, 10, 13
Meta**d**ata **R**epository ..... *See* MDR
methods 8
model 3, 4, 5, 8, 10, 11, 12, 13, 14, 26, 35, 36, 39
Participant 16, 24, 25, 26
Participant Medical Identifier 26
Password 6
Path 5, 16, 17, 26, 29, 30, 39
Permissible Values 25
Projects 12
query 3
Query Wizard 8
Search categories 17
searchable 16
service instances 3, 4, 12, 14
Specimen 16, 17, 26
Specimen Collection Group 26
UML Model ..... *See* Domain Model
User Name 6