caCORE SIW and UML Model Loader

Version 4.0 Technical Guide

Center for Biomedical Informatics
and Information Technology

October 29, 2008

TABLE OF CONTENTS

About This Guide

This chapter provides information about the *caCORE SIW and UML Model Loader Technical Guide* for Version 4.0.

It includes the following topics:

Purpose

The purpose of this guide is to describe the Semantic Integration Workbench (SIW) and UML Loader components of caCORE. caCORE is the cancer Common Ontologic Representation Environment and is an open-source, standards-based, semantics computing environment and tool set created by the National Cancer Institute's Center for Biomedical Informatics and Information Technology (CBIIT).

The SIW provides a way for UML model owners to bind the elements in their models to terms from a controlled vocabulary (EVS concepts). The controlled vocabulary is used to help accurately define the elements in the model. This binding is referred to as "annotation." The purpose of annotating using a controlled vocabulary is to provide ways for disparate models to use the same terms to describe similar model elements that are used in a similar way (such as name or gene or address or patient).

Once the model is annotated with EVS concepts, those concepts are then used to link (where possible) the annotate elements from the model to metadata elements already being used by other UML models registered by other organizations. This linking is referred to as "registration" and is the key to semantically integrating your model with other registered UML models.

The UML Loader is the tool used to register UML model metadata into the caDSR. The caDSR is the Cancer Data Standards Repository, and consists of a database and toolset that are used to create, edit and reuse common data elements. These data elements (also known as DEs or CDEs) represent common pieces of metadata used by registered UML models and which can be re-used by other models as appropriate.

This guide provides an overview of the SIW and the UML Loader and explains how these two components can be used to achieve semantic integration of your UML model with other organizations' models.

Audience

The information in this guide is directed at UML model owners who want to bring their model semantically in line with other similar organizations' systems, and (if desired) to register their system's metadata into caDSR. This guide is also designed to provide other types of readers (non-programmers) with a basic overview of the NCICB's purpose and methods for semantic integration between dispersed research information systems.

This guide also contains information directed at EVS curators who are using the SIW to annotate the XMI files submitted to the NCICB for curation.

While semantic integration typically occurs separately from the use of other caCORE tools (e.g., caAdapter or the SDK), it is a good idea to use the SIW to confirm that the file you are using for the other aspects of developing a caCORE-compatible system is also semantically compatible with other registered caCORE systems' metadata.

This guide provides information about how to use the SIW to bind the elements from any UML model to metadata concepts located in EVS, or to data elements already registered in caDSR.

Topics Covered

The primary information covered in this guide is the use of the SIW by both model owners and EVS curators. This information includes detailed explanations and procedures for using the tool, as well as information on why each step is important.

Because the UML Loader is used exclusively by NCICB personnel, the information included in this guide is limited to an overview of the end result of the UML Loader process. This guide also provides basic instructions for searching the caDSR for your model's loaded elements.

Below is a list of the chapters contained in this guide and a brief overview of the information you will find in each chapter:

- *Chapter 1, Overview of caCORE and Semantic Integration,* on page 5 provides overview information about caCORE and about the purpose and process of semantic integration. It also provides information regarding the steps you must follow before using the SIW.
- *Chapter 2, Using the Semantic Integration Workbench (SIW),* on page 21 provides the basic information you need to know to use the SIW, including an overview of the sections of the SIW viewer and their functionality, and details about the viewer preferences available, how to change the default preferences, and how these changes affect the display of your model information in the SIW viewer.

- *Chapter 3, Semantically Integrating Your UML Model,* on page 41 provides detailed information on how to use the SIW to review and annotate the XMI file exported/saved from your UML modeling software.
- *Chapter 4, caCORE UML Loader and Registering Metadata,* on page 91 provides an overview of the process (including the expected outcome) of registering your model's metadata in caDSR. This chapter also provides basic information on using the caDSR tools to browse for your model's elements once they are loaded.
- *Glossary* on page 121 provides definitions for terms used throughout this guide and for terms that relate to caCORE in general.

Additional References

For more information about caCORE Semantic Integration tools and other caCORE components referenced in this guide, refer to the NCICB web site, which contains information about all of the tools available in the caCORE suite: http://ncicb.nci.nih.gov/

The caCORE Wiki also provides a centralized place for overviews and links to additional information and release notes. The caCORE Wiki can be found at: https://wiki.nci.nih.gov/x/BIAe.

Specific locations on the wiki that may be of interest include:

- **SIW Wiki Page**, which contains links to the Release Notes for the past several releases: https://wiki.nci.nih.gov/x/QYEI.
- **Semantic Integration Tools Project on GForge**, which provides extensive information regarding both past and ongoing development of the SIW and UML Loader: http://gforge.nci.nih.gov/projects/siw/.
- **caDSR Wiki Page**, which contains descriptions of each of the tools in the caDSR toolset, links to release notes, and links to the URLs for each of the available tools: https://wiki.nci.nih.gov/x/OYEI.
- **caCORE SDK Wiki Page**, which contains overview information about the SDK and links to release notes and to the 4.0 documentation suite: https://wiki.nci.nih.gov/x/XIAI.

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 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.

Convention	***Description***	***Example***
text in SMALL CAPS + text in SMALL CAPS	Indicates keys that are pressed simultaneously.	Press SHIFT + CTRL.
Italics	Indicates emphasis on particular words or phrases.	To undo your changes, select a different node *before* clicking **Apply**.
Italics	Used for references to other documents, sections, figures, or tables.	See *Figure 4.5*.
`Italic boldface monospaced 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 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 document.

Semantic Integration Workbench Development and Management Teams		
System and Process Development	***Documentation***	***Project and Product Management***
Christophe Ludet [3]	Bronwyn Gagne [2]	Denise Warzel [1]
Claire Wolfe [4]	Jill Hadfield [1]	Steve Alred [3]
[1] National Cancer Institute Center for Bioinformatics (NCICB)	[2] Lockheed Martin	[3] Oracle
[4] TerpSys		

Contacts and Support	
NCICB Application Support	http://ncicb.nci.nih.gov/NCICB/support Telephone: 301-451-4384 Toll free: 888-478-4423 Email: ncicb@pop.nci.nih.gov

Chapter 1

Overview of caCORE and Semantic Integration

This chapter provides an overview of the NCICB caCORE infrastructure and of the caCORE product release paradigm. This chapter also provides an overview of semantic integration and how it fits into the caCORE process.

The final sections of this chapter provide information and instructions for model owners regarding prerequisites for the UML model prior to exporting it for use with the caCORE Semantic Integration Workbench.

Topics in this chapter include:

- *caCORE Overview* on this page
- *Introduction to Semantic Integration* on page 8
- *Recommended UML Model Pre-requisites* on page 14
- *Generating the XMI file for the SIW* on page 19

The caDSR team maintains a list of commonly asked questions and answers and tips about the SIW and UML Loader on the FAQs - UML Modeling page on the caCORE Wiki at: https://wiki.nci.nih.gov/x/7wFy.

caCORE Overview

The NCI Center for Bioinformatics (NCICB) is an organization that provides biomedical informatics support and integration capabilities to the cancer research community. In order to assist with this integration, the NCICB has developed caCORE, the cancer Common Ontologic Representation Environment (caCORE). caCORE is a data management framework designed for researchers who need to be able to access and provide data across a large number of data sources.

By providing a common data management framework, caCORE helps streamline the informatics development throughout academic, government, and private research labs and clinics. The components of caCORE support the semantic consistency, clarity, and comparability of biomedical research data and information. caCORE is open-source enterprise architecture for NCI-supported research information systems, built using formal techniques from the software engineering and computer science communities.

The four characteristics of a caCORE-compatible system include:

- Model Driven Architecture (MDA)
- n-tier architecture with open Application Programming Interfaces (APIs)
- Use of controlled vocabularies, wherever possible
- Registered metadata

The use of MDA and n-tier architecture, both standard software engineering practices, allows for easy access to data by other applications. The use of controlled vocabularies and registered metadata, less common in conventional software practices, requires specialized tools that are generally unavailable.

As a result, the NCICB (in cooperation with the NCI Office of Communications) has developed the Enterprise Vocabulary Services (EVS) system to supply controlled vocabularies, and the Cancer Data Standards Repository (caDSR) to provide a dynamic metadata registry. The Semantic Integration Workbench (SIW) and UML Loader provide a process-bridge between the EVS, the caDSR and any UML model.

When a system meets all four of the development characteristics listed above, it is said to be “caCORE-like.” Such systems provide several advantages:

- With its open APIs, the end user (whether human or machine) can retrieve data without having to understand the implementation details of the underlying data system.
- The maintainer of the resource can move the data or change implementation details without affecting the ability of remote systems to access the data.
- Most importantly, the system is *semantically interoperable*; it uses runtime-retrievable information to provide an explicit definition and complete data characteristics for each object and attribute supplied by the data system.

Components of caCORE

The components that comprise caCORE include: SDK, SIW, EVS, caDSR, UML Loader, caBIO, CSM, and CLM. The following subsections provide a brief description of each component. You can also find more information on the NCICB web site at: http://ncicb.nci.nih.gov/.

Software Development Kit (SDK)

The caCORE SDK is designed to assist researchers and application developers with creating caCORE-compatible APIs for their system. When use of the caCORE SDK is combined with the use of controlled vocabularies and registered metadata, the resulting software system is considered to be a “semantically integrated caCORE-like system” because the runtime-accessible data elements available through the exposed

APIs will have been defined using controlled terminology. These elements will therefore be easily understood and interpreted by other caCORE-like systems.

Semantic Integration Workbench (SIW)

The SIW is a tool designed to help UML model owners map the elements of their model to metadata concepts already defined in the NCI Thesaurus and, where possible, to data elements already registered in the caDSR. Essentially the SIW provides an easy way to semantically bind the elements from one UML model to terms or definitions used by elements in other systems' models. The ultimate purpose is to see that systems use the same terms for identical items, allowing those systems to more easily gather and/or share data.

Enterprise Vocabulary Services (EVS)

EVS provides controlled vocabulary resources that support the life sciences domain, implemented in a description logics framework. EVS vocabularies provide the semantic raw material from which data elements are constructed.

Cancer Data Standards Repository (caDSR)

The caDSR is a metadata registry based upon the ISO/IEC 11179 standard. It is used to register the descriptive information (metadata) needed to render cancer research data reusable and interoperable. The caBIO, EVS, and caDSR data classes are registered in the caDSR, as are the data elements used on NCI-sponsored clinical trials case report forms, as well as from caBIG- and NCICB-sponsored UML models.

UML Loader

The UML Loader is a tool used by the NCICB to register the metadata elements from a system's UML model into caDSR. The UML Loader helps automate the process of binding UML model metadata elements to registered caDSR components, and where necessary, create new components that correspond to the elements in the model.

Cancer Bioinformatics Infrastructure Objects (caBIO)

The caBIO model and architecture is the primary programmatic interface to caCORE. Each of the caBIO domain objects represents an entity found in biomedical research.

Unified Modeling Language™ (UML) models of biomedical objects are implemented in Java as middleware connected to various cancer research databases to facilitate data integration and consistent representation. Examining the relationships between these objects can reveal biomedical knowledge that was previously buried in various primary data sources.

Common Security Model (CSM)

The Common Security Model (CSM) provides a flexible solution for application security and access control with three main functions:

- Authentication to validate and verify a user's credentials
- Authorization to grant or deny access to data, methods, and objects
- User Authorization Provisioning to allow an administrator to create and assign authorization roles and privileges.

Common Logging Module (CLM)

The Common Logging Module (CLM) provides a separate service under caCORE for audit and logging capabilities. It also comes with a Web-based locator tool.

Client applications can use the CLM directly, without the application using any other components such as the CSM.

About the caCORE Release Paradigm

In September 2007, the NCICB Infrastructure and Product Management Team made the decision to separate the caCORE Components that had previously been bundled and released together. This decision was geared toward allowing each of the infrastructure product teams to be more responsive in addressing specific needs of the user community.

This guide focuses on the 4.0 release of the SIW and UML Loader components of caCORE.

For more details on other caCORE components, refer to the caCORE Overview web page at http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview, which directs you to other product-specific informational pages. You may also want to view the caCORE Wiki (https://wiki.nci.nih.gov/x/BIAe), which provides overviews of the caCORE toolset and links to product-specific documentation.

Introduction to Semantic Integration

Semantic integration refers to the aspect of creating a caCORE-like system that addresses the mapping of data element metadata to controlled vocabularies using concept codes.

For a UML model, proper semantic integration requires that each UML class and attribute gets mapped to appropriate metadata concepts in a controlled vocabulary. At NCICB, the preferred vocabulary is the NCI Thesaurus, maintained by the Enterprise Vocabulary Services (EVS) staff. It is the association of a model's elements with these controlled concepts that allows for unambiguous interpretation of UML model objects and mapping between objects from different domains. The resulting data elements are more sharable and interoperable.

The SIW performs the association of the model's elements using an XMI file exported/ saved from the UML modeling program used to create the model. The mapping of the elements in an XMI file to EVS concepts is called "annotation."

While you can use manual annotation of the UML model to achieve semantic integration, using the Semantic Integration Workbench (SIW) is the preferred method. The SIW streamlines (and partially automates) the process of semantic integration, and required detailed review and approval of each element before a model can be registered into caDSR. Manual annotation has the potential to introduce human errors that can cripple metadata registration and the UML Loader.

Once the annotated version of the model has been approved by the model owner and the EVS curation team, the XMI file is sent to the NCICB caDSR team to be transformed into caDSR metadata via the UML Loader. The caDSR metadata registry, based upon the ISO/IEC 11179 standard, registers the descriptive information needed to render cancer research data reusable and interoperable. For more information about the ISO/IEC 11179 standard, see http://isotc.iso.ch/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm??Redirect=1.

Workflow for Full Semantic Integration

The semantic integration process consists not only of annotating an XMI file and mapping the model elements to other pre-existing elements, but also of adding the appropriate items into the UML model before exporting the XMI file, and then of reloading the mapped items from the annotated XMI file back into the UML model for use in the next version of the system.

While this guide is designed to describe the SIW and the mapping process specifically, it is important that as a model owner you understand the overall process needed to achieve full semantic integration of your UML model.

Semantic integration is divided into four phases:

1. **Phase One** - the model owner creates the necessary tagged values for the model elements, and if necessary, creates any local value domains to be used by the model (local value domains are optional). When the model is complete, the model owner creates an XMI file of the UML model (either by exporting from Enterprise Architect, which creates an .xmi file or saving from ArgoUML, which creates a .uml file).

2. **Phase Two** - using some or all of the five steps in the SIW, the elements in the XMI file are semantically annotated using NCI Thesaurus concepts in an iterative process between the model owner and the EVS curation team. Once the XMI file is fully annotated, the model owner reviews the file and, if accepted, submits the file, along with a submission form, to the NCICB to have the model loaded into the caDSR.

3. **Phase Three** - the NCICB takes the file and the submission request form information and loads the UML model to the caDSR Sandbox using the UML Loader. If the model loads successfully, the model owner reviews and approves the model for loading to Production. Once loaded to Production, the model owner reviews it again and requests any necessary changes to the metadata. After curation, the registered model is sent through compatibility review and once approved, is released to the public.

4. **Phase Four** - after the model has been loaded to caDSR and released, the model owner can use the Roundtrip mode of the SIW to bind the caDSR metadata to the model elements in the XMI file. The model owner can then import all of the XMI file annotations back into the UML model (they are applied to the model as tagged values for each element). This readies the model for reuse or for the next version. Model owners can also produce the final public APIs using the SDK code generator.

At this point you should be starting to understand why semantic integration is useful. If one developer registers their model in caDSR, then a different model owner can compare their model against the registered model and reuse existing elements from the registered model for those same elements in their model. Where the new model has elements not used before, new metadata elements are created for caDSR, which are then available for use by another system's model owner.

Eventually you have a large number and wide range of systems that are using the same metadata for the same objects, making it possible for those systems to communicate and share data with one another more easily than if those systems had been created without using a common vocabulary.

Terms You Should Know

In order to understand the information being provided in this guide, it is important that you understand certain terms as they are used within the context of the caCORE environment and associated toolset. These terms also appear in *Appendix 5, Glossary*, on page 121. However, they are important enough and pervasive enough in this text to be called out separately and defined in relation to semantic integration.

***Metadata* —**Metadata is data about data, or is sometimes is described as the definition of a piece of information. For example, a person's name might be "Joe Smith." The data about the person is "Joe Smith." The metadata for that information is "name" and is defined as "a term that specifically identifies the item to which it is attached."

While "name" can function as a piece of metadata, the term "name" doesn't mean much by itself. The idea of a name can apply to a person, a drug, a facility, or any number of items. It is not the "name" that creates complexity but the attributes that can be associated with "name." If a person is filling out a form, there is probably a place for "First Name" and for "Last Name" and possibly for "Maiden Name." And while these conventions are clear to some users, not all cultures use "first" as "given name" and "last" as "family name."

In order for systems to be able to assemble and share the same sort of information from all participants, they need to be certain that the information exists in their systems in the same way. This is done by using and reusing the same metadata tags for the same information across systems.

The example provided here touches only slightly on the complexity of the terminology used across disparate systems, even if those systems are all working with the same kind of information (i.e., cancer research). Apply the idea of metadata to the collection of information about patient symptoms or drug interactions or physical descriptions and you can begin to understand why using a controlled vocabulary and a centralized metadata repository is important for the sharing of research information.

***UML Model* —**According to Wikipedia, UML or Unified Modeling Language is "a general-purpose modeling language that includes a graphical notation used to create an abstract model of a system, referred to as a UML model." So that basically, a UML model is a graphical representation (picture) of a software application.

Using UML to model your software lets you identify the individual pieces of the application and relate them to one another. The caCORE toolset, including the SIW and the UML Loader is compatible with .xmi files exported from Enterprise Architect (EA) and with .uml files saved from ArgoUML. Throughout this guide, however, both types of files are referred to as XMI files, regardless of file extension.

***XMI*—**XMI stands for "XML Metadata Interchange" and is a standard for exchanging metadata information via Extensible Markup Language (XML). The most common use of XMI is as an interchange format for UML models.

Class—According to Wikipedia, a class is “a programming language construct that is used to group related attributes and methods...a cohesive package of a particular kind of metadata.” For example, your model might contain a class “Patient” which is described in your model as “A person being acted upon by a medical professional.” The details of the object are defined by the attributes for the class (such as name or address or gender). The class is defined in the model, but through semantic integration and the SIW, is also defined using concepts from the EVS controlled vocabulary. The attributes for the class (which are also mapped to EVS concepts) provide further details about the class and are defined separately.

Attribute—An attribute defines the properties of an object. In terms of a UML model, the attributes associated with a class provide details about the use of the class in the system. For example, the model may have a class called “patient” and that class has several attributes associated with it including “gender”, “race”, “ethnicity”, “id” or other descriptors.

Each of the attributes in a model will be mapped to concepts or definitions so that each time they are used in the same way, the same definition follows. For example “id” may be defined as “Identifier - an item that uniquely identifies an object,” and “gender” may be defined as “the assemblage of properties that determines reproductive role.”

Mapping UML model elements to EVS concepts ensures that whenever an object is used in a particular way (even across models), the definition of the object for that use is the same and is unambiguous; it means the same thing every time.

Value Domain—Value domains are defined as the “domain of possible values” for an element in the model. This is typically expressed as the type of information that the system can accept as valid data for an item. For example, your model has an attribute “id” and the system that the model is based on has restricted all IDs to be numeric. The value domain for that attribute might be “java.lang.Integer.” This identifies the type of values that your use of “id” can have.

Value domains can be “enumerated” or “non-enumerated.” Non-enumerated value domains mean that there are restrictions to the possible values for the item but the possible values are not specified. Enumerated value domains mean that the possible values for the item are explicitly stated. So for an attribute like “gender” the value domain might be enumerated and limited to “male” and “female” or to “m” and “f” depending on how the system recognizes the informational input.

Association—An association is simply the relationship between classes in a model, and includes details about that relationship (multiplicity, directionality, etc.). Associations also have “roles” which define how the element relates to the other elements it is associated with.

Tagged Value—A tagged value is a UML construct that represents a name-value pair. It can be attached to anything in a UML model and is used by UML modeling tools to store specific information about the element to which it is attached.

The caCORE toolset uses a variety of tagged values from the UML model to perform a variety of tasks. The SIW in particular uses element description tagged values to identify the purpose or meaning of the element in the model. Specifically, the model owner needs to add tagged values called CADSR_Description to each element in the model to define or describe the element. These descriptions help the EVS curation team properly map the element to EVS concepts. For more information regarding the tagged values used by the SIW and UML Loader, see the XMI Tag Reference wiki page located at: https://wiki.nci.nih.gov/x/SYI8.

***EVS Concepts*—**Each metadata term in the NCI Thesaurus has a code, a name, and a definition for the term as well as an indication of the source of the definition. These are called "concepts" and function as the controlled vocabulary that are mapped to the elements in your UML model. Mapped concepts become tagged values for the elements in the model.

Each element in the model must have at least one concept mapped to it. The first concept mapped to an element is called the Primary Concept. The Primary Concept must reflect the basic meaning or idea of the element. Each concept added after the Primary Concept is called a Qualifier Concept. Each Qualifier Concept mapped to an element provides more detailed information about the element.

For example, your model has an element named "patientBirthDate." The fundamental idea of the element is that it is a date. So the Primary Concept mapped to that element might be the EVS concept called "date" indicating a point in time. The element then might also have two Qualifier Concepts mapped to it called "birth" and "patient." The combination (referred to as Summary Concept) is then identified with the element as "Patient Birth Date" and has three associated concept codes, three concept names and three concept definitions, one for each mapped concept.

This concept mapping or linking through EVS acts as a bridge between the elements in the UML model and the metadata elements in caDSR and provides a common vocabulary for the use and definitions of terms across models.

***Data Element/Common Data Element (DE/CDE)*—**A Data Element (DE) or Common Data Element (CDE) is defined as the unique combination of a Data Element Concept and a Value Domain. The terms *data element* and *common data element* (or the abbreviations DE and CDE) mean the same thing and are used interchangeably.

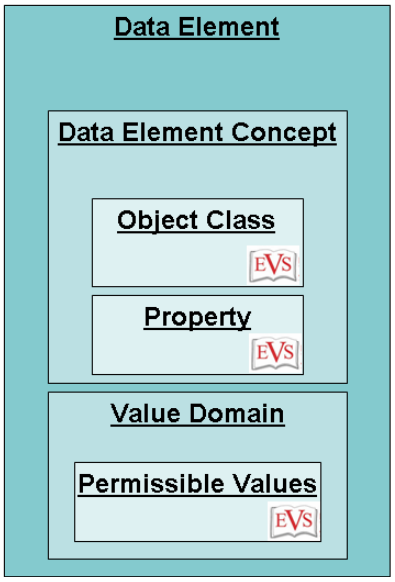


Figure 1.1 Graphical representation of a caDSR Data Element

A Data Element Concept (DEC) is the unique combination of an Object Class and Property. Object Classes correspond to UML model classes, and Properties correspond to UML model attributes. This means that if your UML model has a class that has four attributes associated with it, there will be four DECs created for that item, one for each attribute/class combination.

As each of those DECs is associated with a value domain, an existing data element is reused or a new one is created. So if there are four DECs associated with a class (one for each class/attribute combination), there will be four data elements also associated (when the value domains corresponding to the datatypes associated with each attribute are added to each DEC).

This means that each piece of metadata associated with your model is derived from a combination of the definitions or descriptions established for the item the metadata represents. Remember that each element in your model was defined using one or more EVS concepts, so the full definition and intent of each element is understood, unambiguous and given commonality through the controlled EVS vocabulary. So when the elements of your UML model are mapped to caDSR metadata, you know that the metadata accurately represents the characteristics of the model element (class, attribute, value domain) to which it is mapped.

Graphically, the correlation between UML model elements and the component parts of a DE looks like this:

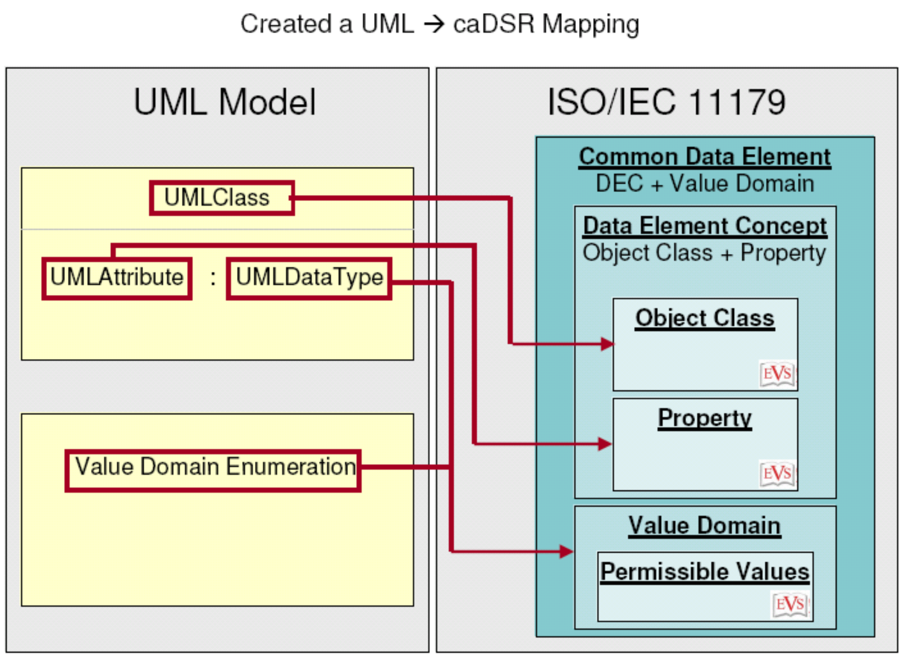


Figure 1.2 Mapping between UML model elements and a caDSR CDE

This mapping or transformation of UML model elements into caDSR metadata is performed initially by the UML Loader, using the concept information provided through the annotation process. After your model has been loaded into caDSR, the Roundtrip mode of the SIW can apply this mapping back into your XMI file for re-loading into the UML model, to use for the next version of the system.

***Unannotated and Annotated XMI Files*—**An Unannotated XMI file is one that has been exported from the UML modeling software but has not yet had all of its elements mapped to either EVS concepts or caDSR CDEs.

When you select the SIW step to Review Unannotated XMI file, the SIW hides concept annotation errors, because by definition, an unannotated file is missing concept mapping. Reviewing an unannotated file allows you to focus on any syntactic errors in the file (elements without definition or descriptions). This allows you to review and (where necessary) change your model before doing the annotation.

An Annotated XMI file is one where all of the elements in the model have been mapped to EVS concepts or to caDSR CDEs. The elements in the file have been annotated with additional information.

When you select the SIW step to Review Annotated XMI file, the SIW shows all errors in the file that will need to be fixed before the model can be loaded into caDSR. These include elements where concept information is missing, where duplicate mapping exists, or where invalid value domains have been used for elements, among other items.

Recommended UML Model Pre-requisites

Since the XMI file you export from your UML modeling software may be used across the caCORE toolset, you want to be sure that the model and the elements within it meet the conventions either recommended or required by the various caCORE tools.

The purpose for this section of this guide is to identify those things that need to be done or verified in the model before exporting the XMI file for use.

The topics in this section include:

- *Modeling Constraints*, below
- *Naming Best Practices* on page 16
- *Using Local Value Domains* on page 17

As a final note, Enterprise Architect (EA) and ArgoUML are the officially supported UML modeling tools for use with the caCORE suite of products. Throughout this guide you will notice the use of XMI to describe the file exported/saved from the UML modeling software and processed by the caCORE tool. XMI is simply the format of the file being processed. It is understood that EA can create an .xmi file and that ArgoUML creates a file with a .uml extension.

Modeling Constraints

Some of the caCORE tools have recommendations and constraints for the UML models to be used with the programs. Those constraints most pertinent to the SIW and UML loader are identified below.

***Constraint 1: Ignored UML Elements*—**While UML class elements are recognized by caCORE tools, and those classes may contain both attributes and operations, operations are ignored by the caCORE tools.

***Constraint 2: Attribute Types*—**Each attribute must have a type assigned to it, and the type must be a Java primitive data type or one of the Java wrapper classes. Each

Java wrapper class must be defined within the model under the appropriate java package (e.g., java.lang.String).

In accordance with object-oriented design principles, attributes of a class that are of a complex type should be modeled as associations. The only exception is if you create a “caDSR Value Domain” stereotyped class in your UML model. For a UML attribute to use this value domain, the model owner must add the tagged value “CADSR Local Value Domain” / “My Value Domain” as described later in *Pointing a UML Attribute to a Local Value Domain* on page 19.

Constraint 3: Recognized Relationships—The caCORE tools recognize association and generalization (inheritance) relationships. In addition, the UML Loader records aggregations and compositions when registering metadata, but for the purposes of generating CDEs, these types of relationships are treated as simple associations.

Constraint 4: Association End Role Names—Both ends of each association should be given a role name (source role and target role).

Constraint 5: Association End Multiplicity—The multiplicity of each association end must be specified at both ends.

Constraint 6: Association End Navigability (Directionality)—The navigability of each association must be specified.

Constraint 7: Packages and the Logical Model—Each model should be placed under the Logical Model package. Classes in the model need to be placed under at least one package under the Logical Model package (for example, add a package named “domain” and place the classes under that). There can be any number of packages under the Logical Model package and you may add them to whatever level depth you choose. Note that all packages under the Logical Model package must conform to Java package limitations (e.g., no spaces).

***Constraint 8: UML Descriptions* —**LL classes and attributes must have textual descriptions associated with them. This means that a definition for each class and each attribute must be entered into the model using the CADSR_Description tagged value. The SIW will present an error for any element that does not contain descriptions for an element. This information is necessary to let the EVS curators know what the purpose of the element is and therefore to properly map it to an EVS concept.

The text provided in the CADSR_Description tagged value is the information that appears in the UML Class Definition or UML Attribute Description field of the SIW. You may have up to eight CADSR_Description tagged values for each element (CADSR_Description, CADSR_Description_2, CADSR_Description_3, etc.). If you have multiple CADSR_Description tagged values, the text is concatenated, up to a maximum of 2000 characters. If you are using EA, you are limited to 256 characters per tag.

Note: The SIW will still recognize the legacy tagged values for classes and attributes (“documentation” and “description” respectively), however the CADSR_Description tagged value is preferred.

Constraint 9: Java Limitations—Because the SDK produces a Java-based system, class and attributes names are subject to the conventions defined by the Java language. UML models should *not* use Java reserved words as class or attribute names (i.e., `int`, `long`, `class`, `interface`, `void`, etc.). Also, model elements must not use hyphens, angle brackets or other reserved characters that will result in Java compilation errors.

Naming Best Practices

To ensure semantic interoperability, you should pay close attention to the class and attribute naming conventions established by the Sun Microsystems Java Bean Specification. Decisions as to the names you apply to your UML element can affect many things, including the automated mapping of the elements to EVS concepts through the Semantic Connector, the mapping of the elements to common data elements through the Roundtrip step, as well as the generation of the system code by the SDK Code Generator.

The following list, while not exhaustive, provides some broad outlines regarding naming best practices that you should consider during modeling.

Adopt a consistent naming convention, using camel case for class and attribute names—Where a term is used and re-used within the model, you should make sure that if the intent and use of the term is the same, that the naming convention and description for the term is also consistent throughout the model. In addition, the Sun Microsystems Java Bean Specification establishes the use of camel case for classes and attributes, and is briefly described below:

- Classes
    - One word class names are capitalized (e.g., “Patient” or “Location”).
    - Multiple word class names should have no space between words and the first and subsequent words are all capitalized. (e.g., SequenceVariant or PatientAddress).
- Attributes
    - One-word attribute names should be all lowercase (e.g., “name” or “gender”).
    - Multiple word attribute names should have no space between words and the first word should be lowercase, while the first letter of second and subsequent words should be capitalized (e.g., streetNumber or initialDiagnosisDate).

The camel case convention is not simply a “nice to have” because it enhances readability; the Semantic Connector step of the SIW parses element names based on camel case, mapping each part of the name it finds to EVS concepts. In this way, using camel case in the model helps to automate the annotation process.

Avoid abbreviations and acronyms—To the extent possible and reasonable, avoid the use abbreviations and acronyms and use full terms instead (e.g., use ‘DatabaseReference’ instead of ‘DbRef’).

Do not use jargon—Avoid the use of ‘jargon’ terms where standard terms exist (e.g., use “microarray” not “chip”).

For the above two items, all you have to do is keep in mind that jargon, abbreviations and acronyms tend to be industry-specific or regionally-based and are not universal.

***Do not repeat class names in attributes* —**Since the attribute is already associated with the class, repeating the class name is not necessary.

For example, for a class element called “Gene” you would use the attribute “name” not “geneName.” Using the latter format causes the Semantic Connector to unnecessarily map the concept code(s) for “Gene” multiple times for a single element, which is semantically undesirable. While any repetitive naming should be caught during the curation process, it is good practice to keep these out of the model in the first place.

***Do not use Java reserved words as class or attribute names*—**Using Java reserved words prevents the SDK Code Generator from compiling the generated system code. This is also stated as *Constraint 9: Java Limitations* on page 15.

Using Local Value Domains

Most model owners use existing caDSR value domains to map to their attributes, and and you are strongly encouraged to use caDSR value domains whenever possible. However, you may create local value domains (LVD) within your model by creating classes and stereotyping them with the proper tagged values (as noted in this section).

In order to use LVDs for attribute mapping, you must create the value domains in the model, apply value meaning annotations to the value domains, and annotate the appropriate attributes with the intended value domain mapping before exporting the XMI file. The SIW cannot be used to map attributes to LVDs.

Note: If you are using LVDs, when your model is submitted for registration into caDSR, the UML Loader will try to match the LVD to an existing value domain with the same long name and permissible values, or create new value domains where there is no match.

You also have the option to map an LVD to a caDSR value domain. Doing so will make reloading easier because the UML Loader will no longer try to find a match based on long name. However the value domain in caDSR will not be changed in any way to reflect any changes to the LVD that might be in your model.

Creating Local Value Domains

For the UML Loader to create an LVD for use with the model’s attributes, the model owner must first create a UML class in the model and stereotype it as a “CADSR Value Domain” or an “enumeration.” When a UML class is stereotyped as “CADSR Value Domain” the SIW identifies it as a value domain rather than an Object Class. The name of the stereotyped class will become the name of the value domain in caDSR, unless you map the class to EVS concepts, in which case the class will take on the naming convention mapped through those concepts.

caDSR naming conventions specify that the representation type should be the last term in the value domain name, such as in “State Code” where the representation term is “Code” and not “State.”

LVDs should be created in a separate package from the regular domain objects.

The following six tagged values must be present in the “CADSR Value Domain” or the “enumeration” class in order to be recognized as a valid value domain.

- **CADSR_ValueDomainDefinition** - Used as the Value Domain Preferred Definition.
- **CADSR_ValueDomainDatatype** - Used as the underlying datatype for the value domain and must be one of the valid caDSR datatypes.
- **CADSR_ValueDomainType** - Used as the underlying value domain type and must be one of 'E' (for Enumerated value domains) or 'N' (for Non-enumerated value domains).
- **CADSR_ConceptualDomainPublicID** - The combination of conceptual domain public id and version must point to an existing conceptual domain in the caDSR.
- **CADSR_ConceptualDomainVersion** - The Conceptual Domain tagged value for this class should be entered as either a whole number or a decimal e.g. 1 or 1.0 for “Version 1.0”.
- **CADSR_RepresentationPublicID** - The Representation Public Id tagged value for this class should be entered as the public id of an existing caDSR Representation Term.
- **CADSR_RepresentationVersion** - the Representation Version tagged value for this class should be entered as either a whole number or a decimal e.g. 1 or 1.0.

The list of optional tags listed below are used to indicate a “top level” concept for a value domain. This is also referred to as a “referenced value domain.”

In a non-enumerated (N) value domain, the reference to a top level concept indicates that the values are not explicitly listed as attributes of the value domain class, but that data values for the attribute are constrained to children of the top level concept.

When a top level concept is present for an enumerated domain (E), this indicates that the permitted data values are listed explicitly as attributes of the value domain class and are children of the top level concept.

The following is a list of the optional tagged values for value domains:

- ValueDomainConceptCode
- ValueDomainConceptDefinition[_n]
- ValueDomainConceptDefinitionSource
- ValueDomainConceptPreferredName
- ValueDomainQualifierConceptCodeN
- ValueDomainQualifierConceptDefinitionN[_n]
- ValueDomainQualifierConceptDefinitionSourceN
- ValueDomainQualifierConceptPreferredNameN

For more information on mapping value domains in caDSR, see *Mapping a UML Class to a Value Domain* on page 104

Value Meanings

Each attribute within a “CADSR Value Domain” or “enumeration” class is considered a “permissible value.” Mapping concepts to those attributes creates “value meanings.” Adding permissible values (attributes) to a value domain class and then mapping them to concepts is the preferred way to create value meanings. Annotation for permissible values is done the same way as it is for attributes with the exception that the tag names are different:

- ValueMeaningConceptCode
- ValueMeaningConceptDefinition[_n]
- ValueMeaningConceptDefinitionSource
- ValueMeaningConceptPreferredName
- ValueMeaningQualifierConceptCodeN
- ValueMeaningQualifierConceptDefinitionN[_n]
- ValueMeaningQualifierConceptDefinitionSourceN
- ValueMeaningQualifierConceptPreferredNameN

The value meaning name is created by concatenating the concepts mapped to the permissible value. The value meaning description is created by concatenating the concept definitions mapped to the permissible value.

Pointing a UML Attribute to a Local Value Domain

In order to indicate that a UML attribute should use a value domain defined within the model, the model owner should add a tagged value of type “CADSR Local Value Domain” to the attribute in the model. The value is the name given to the LVD.

For example, the model owner has created a class that has been stereotyped “CADSR Value Domain” with a name of “MyValueDomain.” For a UML attribute to use this value domain, the model owner adds a tagged value called “CADSR Local Value Domain” / “MyValueDomain” to the attribute.

Again, this must be done in the model *before* exporting the XMI file for processing through the SIW.

Generating the XMI file for the SIW

The SIW uses a pre-processor called the XMI Handler to extract the XML tags needed to perform a semantic annotation of UML classes, attributes and associations from the UML model. It also extracts various other model features and any caCORE specific tagged values that were entered by the model owner.

If you are exporting from EA, you must be sure that the DTD option is *not* checked and that the EA Roundtrip option *is* checked. The extension for the exported file must be `.xmi`.

If you are using ArgoUML as your modeling tool, save your model as a file with a `.uml` extension.

Note: The “Fix XMI” task is no longer needed to prepare the file for use with SIW or caAdapter. Therefore the XMI file can be imported and exported from your modeling tool to make changes without losing any existing SIW concept annotations.

CHAPTER 2
USING THE SEMANTIC INTEGRATION WORKBENCH (SIW)

This chapter provides information regarding how to open the Semantic Integration Workbench (SIW), an overview of the options available in the SIW, how to navigate the SIW Viewer, and how to customize the viewer to your personal viewing preferences. The last sections of this chapter provide instructions for saving changes to your XMI file and for updating your UML model with the changes made through the SIW.

Topics in this chapter include:

Introduction

The Semantic Integration Workbench (SIW) is designed to help UML model owners work through the semantic annotation process, and to remove (whenever possible) the need for users to understand the more complicated details of semantically integrating their model to other models registered in the caDSR. The caDSR is a metadata repository managed by the NCICB that is designed to provide consistently managed metadata for use by other research systems and data repositories.

The SIW essentially helps UML model owners bring the metadata for their models into line with the models used by other facilities. To do this, the SIW contains the following capabilities:

- Automates the searching for and (where possible) the matching of UML elements to items already resident in a controlled vocabulary (EVS).
- Streamlines semantic annotation by offering direct queries to the NCI Thesaurus, inserting the concept information from the search into the user's file in the SIW, creating the appropriate tag names automatically. This eliminates syntax errors in the final XMI file.
- Allows for curation of the XMI file which maps existing or new EVS concepts (not yet in NCI Thesaurus) to classes and attributes. (Curation is performed by NCICB personnel.)
- Allows for mapping of attributes to existing caDSR data elements, either automatically through the Roundtrip step or manually through the SIW viewer.
- Allows review of the final XMI file before it is loaded, providing for individual review and acceptance of each entry.
- Ensures that files submitted for loading into the caDSR have been checked for missing information and validated using Silver Level Compatibility rules.

Pre-Requisites to Using the SIW

Using the SIW requires that you have a valid XMI file exported from either Enterprise Architect (EA) or ArgoUML and that you have the Java Web Start application installed on your computer. For more information, see the sections that follow.

Note: Throughout this guide, the term "XMI" refers to the type of file output from either your UML modeling software or the SIW itself. If you are using Enterprise Architect, the output file will have an .xmi extension; if you are using ArgoUML, the output file will have a .uml extension. In either case, these are referred to as XMI files.

Possessing a Valid XMI File

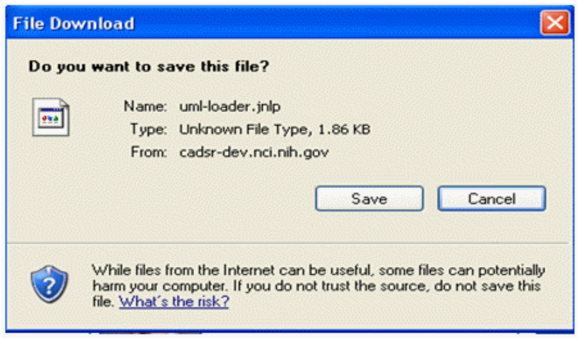
Before you can use the SIW, you need to have a valid file for the SIW to work with. If you are starting with an un-annotated file, be sure you have a valid export of an .xmi file from EA or have saved a .uml file from ArgoUML. If you have not generated this file yet, see *Generating the XMI file for the SIW* on page 19. If you are using EA and your file was not exported using the appropriate options, the SIW will not be able to parse the file properly for use.

Note: The "Fix XMI" task (used in previous versions) is no longer needed to prepare the file for use with SIW. Therefore the XMI file can be imported and exported from your modeling tool to make changes without losing SIW concept annotations.

Installing/Verifying Installation of Java Web Start

The SIW is a Java Web Start application. If Java Web Start is installed on your computer, the SIW will launch when you open the SIW URL. SIW always to checks to be sure you are using the most current version of Java Web Start. If you are not, the SIW automatically retrieves the update for you.

If you do not have Java Web Start installed, you will receive a File Download box when you attempt to access the SIW URL.


File Download

Do you want to save this file?

Name: uml-loader.jnlp
Type: Unknown File Type, 1.86 KB
From: cadsr-dev.nci.nih.gov

While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not save this file. What's the risk?

Figure 2.1 SIW File Download dialog box

If you receive this dialog box, it means that the SIW did not find the application needed to open the SIW file. Click **Cancel**, and go to the Sun web site (http://java.sun.com) to download and install Java Web Start.

Launching the SIW

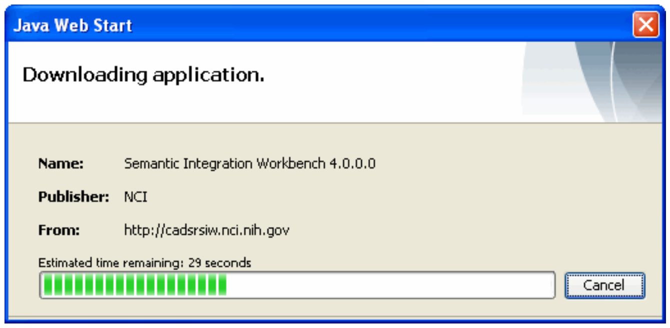
Once you have a valid UML model file to use, and that you have Java Web Start installed, you can begin using the SIW.

The instructions in this section and throughout this chapter use the “browser-launch” method of accessing the SIW. This is done purely for convenience and you do not need a browser to access the SIW. If you prefer, you may use any of the following methods:

- On UNIX or LINUX, type **`javaws`** at the command prompt
- On Windows, launch the JavaWebstart application (the program may appear in your Start menu)
- Save the `siw.jnlp` file to your computer and double-click it to launch the SIW.
- Open the SIW URL: http://cadsrsiw.nci.nih.gov/.

To launch the SIW:

1. Access the following URL using your browser: http://cadsrsiw.nci.nih.gov/.
2. The Java Web Start dialog box appears, showing the Semantic Integration Workbench start-up progress. If you have never used the SIW or not used it in some time, this may take a few minutes.

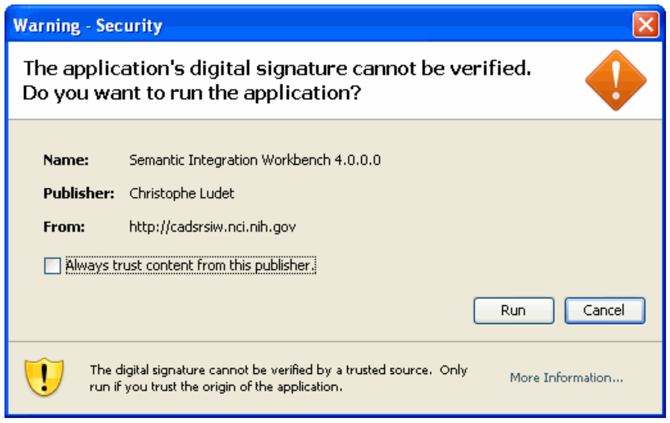


Downloading application.

Name: Semantic Integration Workbench 4.0.0.0
Publisher: NCI
From: http://cadsrsiw.nci.nih.gov
Estimated time remaining: 29 seconds

Figure 2.2 Java Web Start dialog box and progress bar

3. If you receive a Security Warning dialog box, enable the **Always trust content from this publisher** checkbox and then click **Run**.



The application's digital signature cannot be verified.
Do you want to run the application?

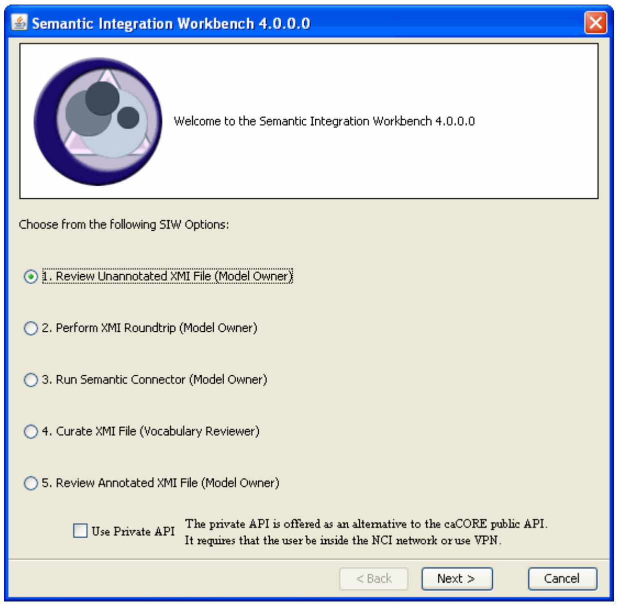
Name: Semantic Integration Workbench 4.0.0.0
Publisher: Christophe Ludet
From: http://cadsrsiw.nci.nih.gov

The digital signature cannot be verified by a trusted source. Only run if you trust the origin of the application.

Figure 2.3 Security Warning dialog box requesting approval to run the application

Note: During installation, portions of the application are downloaded to cache, but the entire application itself is not downloaded.

After the start-up process completes, the SIW opens, starting with the SIW Welcome screen.


Semantic Integration Workbench 4.0.0.0

Welcome to the Semantic Integration Workbench 4.0.0.0

Choose from the following SIW Options:

- [✓ 1. Review Unannotated XMI File (Model Owner)]
- 2. Perform XMI Roundtrip (Model Owner)
- 3. Run Semantic Connector (Model Owner)
- 4. Curate XMI File (Vocabulary Reviewer)
- 5. Review Annotated XMI File (Model Owner)

☐ Use Private API — The private API is offered as an alternative to the caCORE public API. It requires that the user be inside the NCI network or use VPN.

Figure 2.4 Semantic Integration Workbench Welcome Screen

The top of the SIW Welcome screen shows which version of the SIW you are working with (it should be the latest release version). The main portion of the screen lists the five options or steps for using the SIW.

For the most part, the SIW steps should be performed in the order in which they appear on the Welcome screen. However, some of the steps are optional and only apply if you are working with a new version of a previously loaded XMI file, or if you intend to have your model uploaded to the caDSR.

The options available through the SIW include:

Review Unannotated XMI File (Model Owner)—This step allows the model owner to view the XMI representation of their UML model. It provides an easy way to check the model for missing object definition tags or other problems with the file that will require changes before continuing. This step should be performed with all newly exported files before running any other SIW options.

When you view an XMI file in “Unannotated” mode, the errors listed include only syntactic errors, such as missing element descriptions. This allows you to easily see where additional information must be added to or changed in the model in order for the SIW to successfully annotate the model (map to EVS concepts or caDSR data elements) in later steps.

See *Reviewing an Unannotated XMI File* on page 42 for more information on performing this step.

Perform XMI Roundtrip (Model Owner)—This step automatically annotates the XMI file (or updates existing annotations) based on another previously loaded model. The Roundtrip task can save model owners a considerable amount of time because it automatically maps the XMI file’s attributes to existing caDSR common data elements

(CDEs). The automated matching is based on the new model having used exactly the same attribute names as a previous model.

See *Using the XMI Roundtrip Mode* on page 48 for more information on performing this step.

Run Semantic Connector (Model Owner)—This step launches the Semantic Connector, which performs an EVS search against the name of each element in the XMI file (using recognition of camel case to identify those elements with multiple parts). When matches are found, the SIW attaches one or more EVS concepts to each element. This process produces an annotated XMI file (with “FirstPass” appended to the front of the file name) that can then be reviewed by the owner (if desired) before being submitted for curation.

See *Running the Semantic Connector* on page 54 for more information on performing this step.

Curate XMI File (Vocabulary Reviewer)—This step is performed by the EVS concept curation team. During this step, the EVS team uses new or existing EVS concepts to annotate the model, and adds and removes concepts as they determine necessary. The curated XMI file is then returned to the model owner for review and final approval.

This and the next step of the SIW process are typically the most iterative, meaning that very often, the file is returned to the curation team after model owner review, and then back to the model owner after re-curation and verification.

See *Curating XMI Files* on page 58 for more information on this process.

Review Annotated XMI File (Model Owner)—This step, along with the Curate XMI File step (above) will likely be performed several times until all of the model elements are mapped properly. This step allows model owners or model reviewers to review element annotations, search EVS for concepts, and where necessary, change the mapping between a class or attribute and their EVS concept(s). Users may also choose to map annotated elements to existing caDSR CDEs and/or value domains.

This mode of the SIW also provides the ability to run validation checks to ensure that the concept information in the XMI file corresponds with EVS concept information, or that any differences between the XMI file and EVS are determined (by the model owner) to be appropriate and valid.

After curation and review, the model owner can “verify” each element in the file. Once all elements have been verified by the model owner, the file is considered to be complete. The final outcome of this step is a fully annotated XMI file that can be used to register the model into caDSR, and which can also be used as input to the next version of the model.

See *Reviewing an Annotated XMI File* on page 74 for more information on performing this step.

Use Private API —At the bottom of the Welcome screen is a checkbox that allows you to select to use a Private API for SIW processing. Using the Private API performs somewhat faster than using the public API, but you must be connected to the local NCI LAN via VPN or from behind the firewall to use it.

Package Filtering—All of the SIW options can be run on a full XMI file, or you may choose to run each step on only selected elements of the file. When you identify the file to parse, you may select to **Choose Classes and Packages**. Selecting this option adds a screen to the processing wizard that lists the Packages within the XMI file. Expanding a Package shows the classes contained within that package and lets you identify only those you want to process for that step.

Using the SIW Viewer

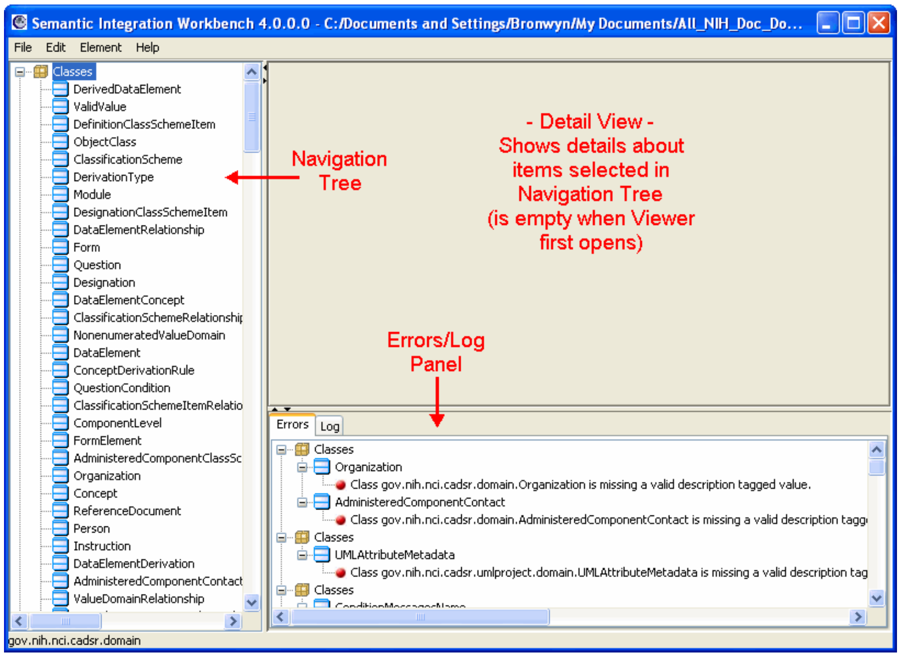
The SIW viewer is the window that appears whenever you select to review or curate your XMI file. The viewer shows the model elements parsed from your XMI file and details about each of those elements. What details you see depends on several things, including the current state of your XMI file (unannotated, annotated, curated). However, the basic attributes of the SIW viewer remain the same, regardless of where you are in the semantic integration process.

This section discusses navigation as well as the functionality available through the SIW viewer, including setting Viewer Preferences. Since the viewer is used to both review and make changes to your XMI file, detailed information regarding how to use specific functionality in the viewer resides in the sections that discuss the steps in which those actions should take place.

Topics in this section include:

- *Navigation Tree* on page 28
- *Detail View* on page 29
- *Errors/Log Panel* on page 34
- *Setting Viewer Preferences* on page 36.

The SIW viewer window has three sections: the navigation tree, the detail view and the errors/log panel.



Class gov.nih.nci.cadsr.domain.Organization is missing a valid description tagged value.

Class gov.nih.nci.cadsr.domain.AdministeredComponentContact is missing a valid description tagg

Class gov.nih.nci.cadsr.umlproject.domain.UMLAttributeMetadata is missing a valid description tag

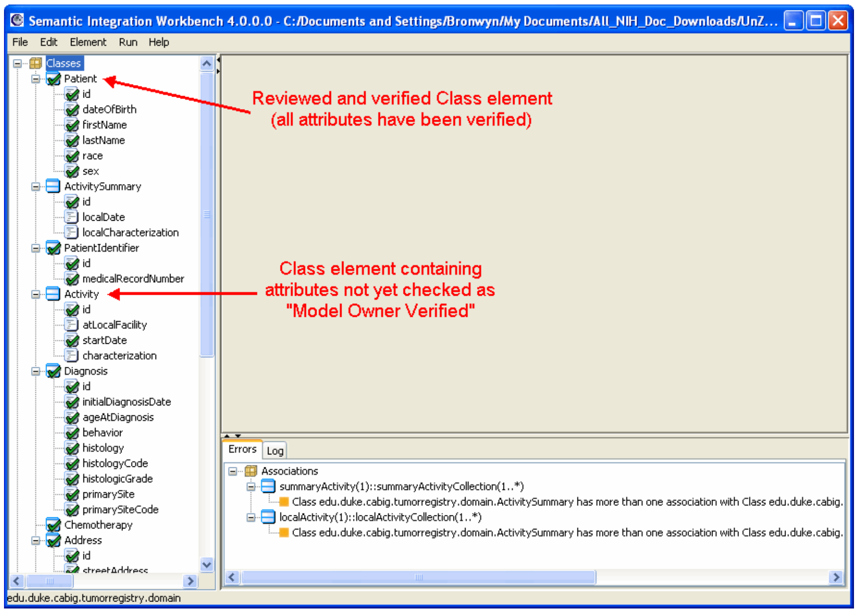
Figure 2.5 SIW Viewer with sections labeled

As is typical of "Windows Explorer-like" applications, you can change the size of the sections of the window by hovering your mouse over the border of that section until the mouse cursor becomes a two-way arrow, and then clicking and dragging the border in the direction of the change you want to make.

Navigation Tree

The navigation tree, located on the left side of the viewer, is a tree-view of all of the elements resident in the model (or packages if only a portion of the file was processed). Selecting an element from the Tree opens a tab in the detail view of the window, showing detailed information for the selected item.

If you are viewing a curated or annotated file, the navigation tree may include checkmarks on elements in the file indicating that the mappings for those elements have been reviewed and checked as “Human Verified” or “Model Owner Verified.”



```
Associations
  summaryActivity(1)::summaryActivityCollection(1..*)
    Class edu.duke.cabig.tumorregistry.domain.ActivitySummary has more than one association with Class edu.duke.cabig.
  localActivity(1)::localActivityCollection(1..*)
    Class edu.duke.cabig.tumorregistry.domain.ActivitySummary has more than one association with Class edu.duke.cabig.
```

Figure 2.6 Navigation Tree showing fully verified and partially verified classes

When the viewer first opens, all nodes of the tree are expanded to display all internal nodes, however each node can be expanded or collapsed by clicking the **+** (plus) or **-** (minus) sign to the left of the node.

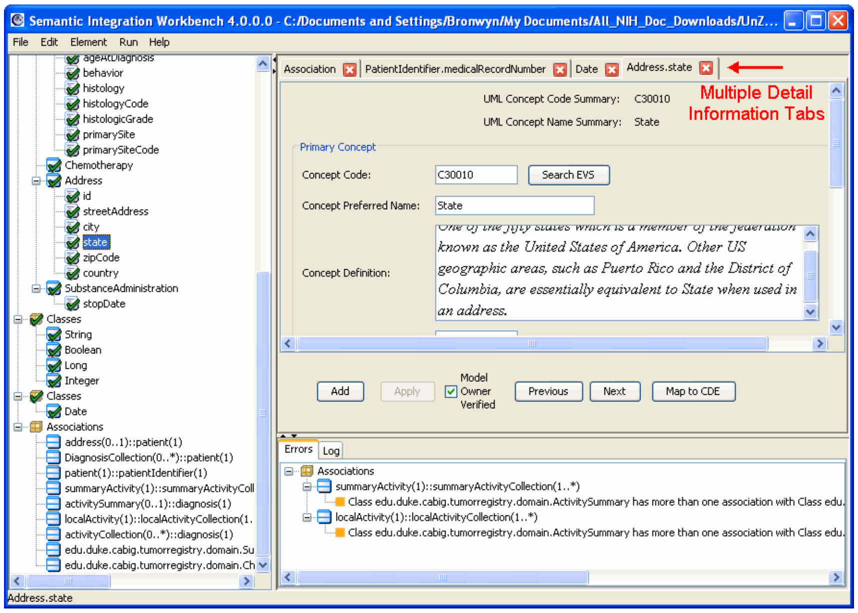
By default, navigation tree items are listed in the order of the structure of your model as indicated in your XMI file. In addition, classes are listed first, with associations appearing at the bottom of the tree. You can change these aspects of the navigation tree display using SIW Viewer Preferences (available through the Edit menu in the viewer). See *Setting Viewer Preferences* on page 36 for more information.

Detail View

The detail view is located in the top-right section of the viewer and is blank when the viewer first opens. Highlighting an element in the Navigation tree opens a tab for that item in the detail view. What information appears in the tab depends on what type of item is selected (class, attribute or association) and the current state of the file you are working with (annotated, unannotated, curated, reviewed, etc.).

For example, if you click on a class or attribute node, the detail view may show only UML Class Documentation text or it may show concept and value domain information, depending on whether or not the selected item has been annotated with concept information yet.

Navigation Note—The detail view can contain multiple tabs at any given time, and will open a new tab for different types of elements selected in the navigation tree. The title on each tab corresponds to the node information it contains. To close a tab, click the “X” located to the right of the node name on the tab.


Semantic Integration Workbench 4.0.0.0 - C:/Documents and Settings/Bronwyn/My Documents/All_NIH_Doc_Downloads/UnZ...
File Edit Element Run Help
ageAtDiagnosis
behavior
histology
histologyCode
histologicGrade
primarySite
primarySiteCode
Chemotherapy
Address
id
streetAddress
city
state
zipCode
country
SubstanceAdministration
stopDate
Classes
String
Boolean
Long
Integer
Classes
Date
Associations
address(0..1)::patient(1)
DiagnosisCollection(0..*)::patient(1)
patient(1)::patientIdentifier(1)
summaryActivity(1)::summaryActivityColl
activitySummary(0..1)::diagnosis(1)
localActivity(1)::localActivityCollection(1.
activityCollection(0..*)::diagnosis(1)
edu.duke.cabig.tumorregistry.domain.Su
edu.duke.cabig.tumorregistry.domain.Ch
Association
PatientIdentifier.medicalRecordNumber
Date
Address.state
Multiple Detail Information Tabs
Primary Concept
Search EVS
Add
Apply
Model Owner Verified
Previous
Next
Map to CDE
Errors
Log
Address.state

UML Concept Code Summary: C30010

UML Concept Name Summary: State

Concept Code: C30010

Concept Preferred Name: State

Concept Definition: One of the fifty states which is a member of the federation known as the United States of America. Other US geographic areas, such as Puerto Rico and the District of Columbia, are essentially equivalent to State when used in an address.

```
Associations
  summaryActivity(1)::summaryActivityCollection(1..*)
    Class edu.duke.cabig.tumorregistry.domain.ActivitySummary has more than one association with Class edu.
  localActivity(1)::localActivityCollection(1..*)
    Class edu.duke.cabig.tumorregistry.domain.ActivitySummary has more than one association with Class edu.
```

Figure 2.7 SIW Viewer with multiple tabs open in the Detail View

Besides definition and/or concept information about the node selected, the detail view also contains buttons that perform a variety of tasks. Some buttons, like **Map to CDE**, do not appear until the file and/or the node selected has reached a certain level of annotation. Other buttons always appear and function as follows:

- **Previous** - Changes the selection in the navigation tree to the item immediately preceding (above) the currently displayed element.
- **Next** - Changes the selection in the navigation tree to the item immediately following (below) the currently displayed element.
- **Add** - Opens blank fields for entering concept information, allowing you to add a Primary Concept or Qualifier Concept (if a Primary Concept is already mapped) to the item.
- **Search EVS** - Opens a search box allowing you to search the NCI Thesaurus for concept information to add or update for the item. The **Search EVS** button always appears next to concept code field, either for existing mapped concepts or as a function of clicking the **Add** button (to populate blank concept information fields).
- **Apply** - Applies the changes made to the concept information. The Apply button is disabled until all concept fields are populated for newly added concepts, or until a change is made to existing concept information.

  You must click **Apply** before navigating away from the changed item’s node, or your changes will be discarded. The SIW prompts you to apply your changes if you attempt to view another node before clicking **Apply**.

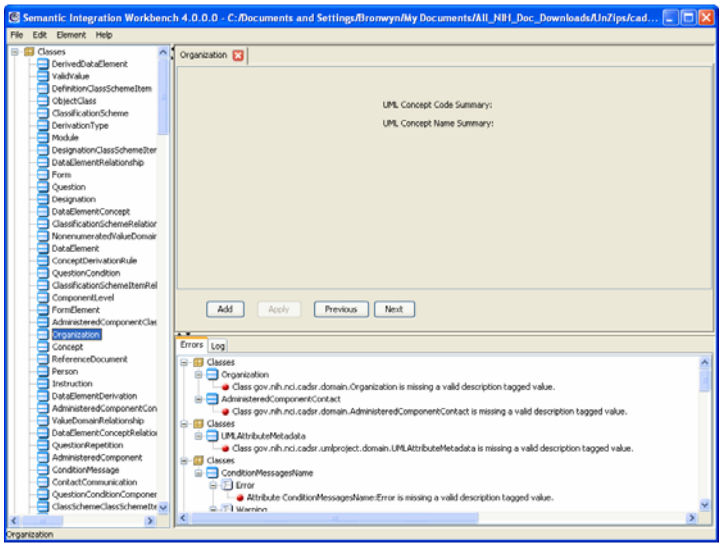
Important: Clicking **Apply** is *not* the same as saving the file. Apply simply applies your changes to the currently open session. You must still save the file (**File > Save** or **File > Save As**) to create an updated XMI file containing the applied changes.

The following sections describe what information appears in the detail view and how that information is displayed, depending on the type of node selected and the information available for the selected item.

Viewing an Unannotated XMI File

An "unannotated" XMI file is one where none or some of the classes and attributes in the file have been annotated with EVS concepts. If you are working with a new version of a new system that has never been through the caCORE semantic integration process, it is likely that your file contains no annotation. If a previous version of your UML model has been through semantic integration, some (even most) of your model elements may be mapped to either EVS concepts or to caDSR data elements, however there are likely new elements that need to be annotated.

When viewing an unannotated file, if you select a class or attribute from the navigation tree and the detail view remains empty (no information appears), this indicates that the element does not have a description tag in the model. Model owners must provide descriptions for each element in the model.



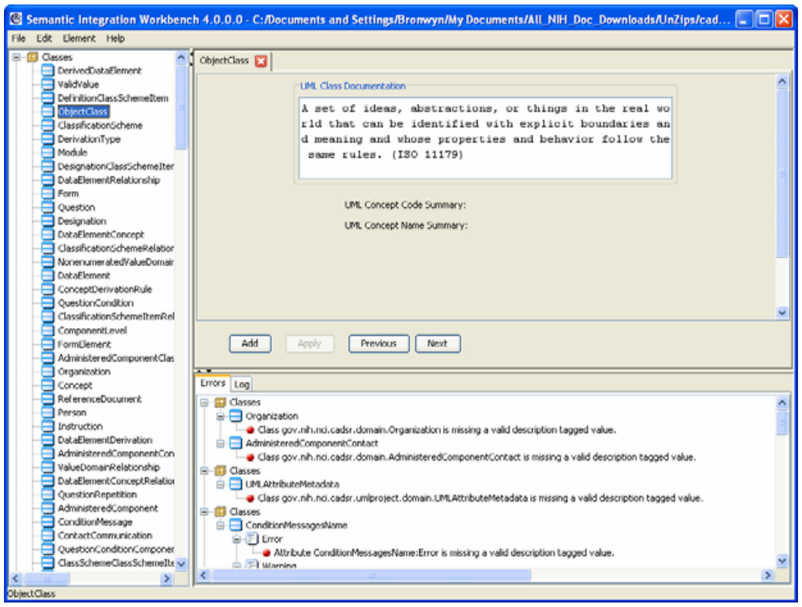
```
Classes
  Organization
    Class gov.nih.nci.cadsr.domain.Organization is missing a valid description tagged value.
  AdministeredComponentContact
    Class gov.nih.nci.cadsr.domain.AdministeredComponentContact is missing a valid description tagged value.
Classes
  UMLAttributeMetadata
    Class gov.nih.nci.cadsr.umlproject.domain.UMLAttributeMetadata is missing a valid description tagged value.
Classes
  ConditionMessagesName
    Error
      Attribute ConditionMessagesName:Error is missing a valid description tagged value.
```

Figure 2.8 Class element selected, with no concept information or UML definition

UML element descriptions are required because they allow EVS curators understand the model owner's purpose and intent for the element. This provides for more accurate mapping of the element to existing EVS concepts and for easier creation of new concepts when applicable.

***A note about UML element descriptions*—**The description tagged values in the UML model should describe the element to which they are attached. Model owners can now use a tag called **CADSR_Description** to tag these elements in their model. If this tag is present, the SIW automatically uses this information to populate the UML element description field. This allows model owners to differentiate caDSR-specific descriptions from other descriptors used for the element in the model. If the model element does not have any CADSR_Description tags, the other existing description tags are used (as previous versions did) to provide a description for the element.

If you select a class or attribute node and a UML description appears, but there is no concept information, this means that the element contains a description in the UML model, which has been exported into the XMI file, however the item has not yet been mapped to an EVS concept.



ObjectClass

UML Class Documentation

```
A set of ideas, abstractions, or things in the real wo
rld that can be identified with explicit boundaries an
d meaning and whose properties and behavior follow the
 same rules. (ISO 11179)
```

UML Concept Code Summary:

UML Concept Name Summary:

```
Classes
  Organization
    Class gov.nih.nci.cadsr.domain.Organization is missing a valid description tagged value.
  AdministeredComponentContact
    Class gov.nih.nci.cadsr.domain.AdministeredComponentContact is missing a valid description tagged value.
Classes
  UMLAttributeMetadata
    Class gov.nih.nci.cadsr.umlproject.domain.UMLAttributeMetadata is missing a valid description tagged value.
Classes
  ConditionMessagesName
    Error
      Attribute ConditionMessagesName:Error is missing a valid description tagged value.
    Warning
```

Figure 2.9 Class element selected showing a UML definition but no concept information.

From this point, you can manually add concept information to the element (by clicking **Add**), or wait to process the file through the Semantic Connector to have the element mapped for you (using the closest match to the element name found in EVS).

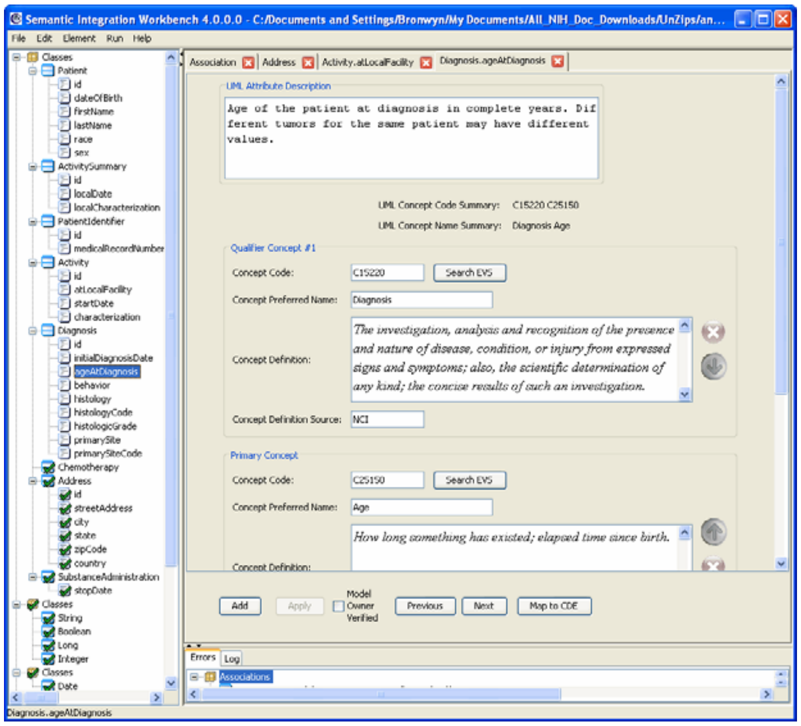
Viewing an Annotated XMI File

An "annotated" XMI file is one where all of the elements in the file have been mapped to either EVS concepts or directly to caDSR data elements. If you select a class or attribute from the navigation tree and the detail view displays concept information for the element, this means that the class or attribute has been mapped to one or more concepts in EVS or "annotated."

By default, concept information appears in "semantic order" meaning that it appears in an order consistent with how the concept names will be positioned when forming the name of the element in caDSR.

For example, if the element is a class titled "Date" it will likely have a single concept mapped to it – a Primary Concept called "Date." However, if the element is an class of the Date class titled "diagnosisDate" then the element will likely have two concepts mapped to it – a Primary Concept called "Date" and a Qualifier Concept called "Diagnosis."

In this example, the detail view will show the Qualifier Concept first and the Primary Concept second because that is how the element is to be named: "diagnosisDate."



UML Attribute Description

```
Age of the patient at diagnosis in complete years. Dif
ferent tumors for the same patient may have different
values.
```

UML Concept Code Summary: C15220 C25150

UML Concept Name Summary: Diagnosis Age

Qualifier Concept #1

Concept Code: C15220

Concept Preferred Name: Diagnosis

Concept Definition: *The investigation, analysis and recognition of the presence and nature of disease, condition, or injury from expressed signs and symptoms; also, the scientific determination of any kind; the concise results of such an investigation.*

Concept Definition Source: NCI

Primary Concept

Concept Code: C25150

Concept Preferred Name: Age

Concept Definition: *How long something has existed; elapsed time since birth.*

Figure 2.10 Attribute element selected, showing mapped concept information

The detail view shows the original UML definition at the top, followed by the Qualifier Concept items (if there are any) with the Primary Concept below that. The value domain (if mapped) is listed at the bottom. Notice that the Concept Summary Code and the Concept Summary Name are the codes and names of each of the mapped concepts in semantic order (qualifier(s) + primary = concept name). This Concept Summary Name should match or closely match the actual name of the element.

If necessary, you can change the order in which the concept information appears in the detail view using options available in the Viewer Preferences. See *Setting Viewer Preferences* on page 36 for more information.

Navigation Note—The concept information section of the detail view will likely scroll (and may scroll significantly) because it contains more information than can be displayed at once. *Figure 2.10* above shows an expanded detail view, in order to display as many pieces of the concept information as possible. However, if you cannot see concept information in the detail view for an element, scroll to see if the information lies above or below the portion of the view being shown.

The detail view also contains additional buttons located to the right of the concept definition fields: up and down arrows and an “X” button. More detailed information about these buttons appears in *Changing Concept Information in Review Mode* on page 78.

Viewing Associations

By default, associations appear at the bottom of the navigation tree. If you select an association node from the navigation tree, four tabs appear in the detail view: Detail, Role, Source, and Target. The elements defined for an association (the Role, the Source and the Target) can be annotated with concepts just like other elements in the model, although this is not required for the file to be considered fully annotated.

The **Detail** tab shows the connection details for the association, including direction, source class, source role, source multiplicity, target class, target role and target multiplicity.

If the associated elements are annotated with concept information, the tabs corresponding to those elements display that concept information. Those tabs also contain the same buttons as other element detail tabs (**Previous**, **Next**, **Add**, **Apply**, and **Search EVS** where applicable). This allows you to manually map those association elements to EVS concepts if desired.

While you can manually map an association's elements to EVS concepts, you cannot change an association's connection information through the SIW.

Errors/Log Panel

Located in the bottom-right section of the viewer, the errors/log panel provides two tabs containing information about the file currently open in the viewer. The Errors tab lists all errors and warnings flagged during processing of the XMI file.

Located behind the Errors tab in the bottom-right section of the viewer is the Log tab, which provides a log of the file parsing process.

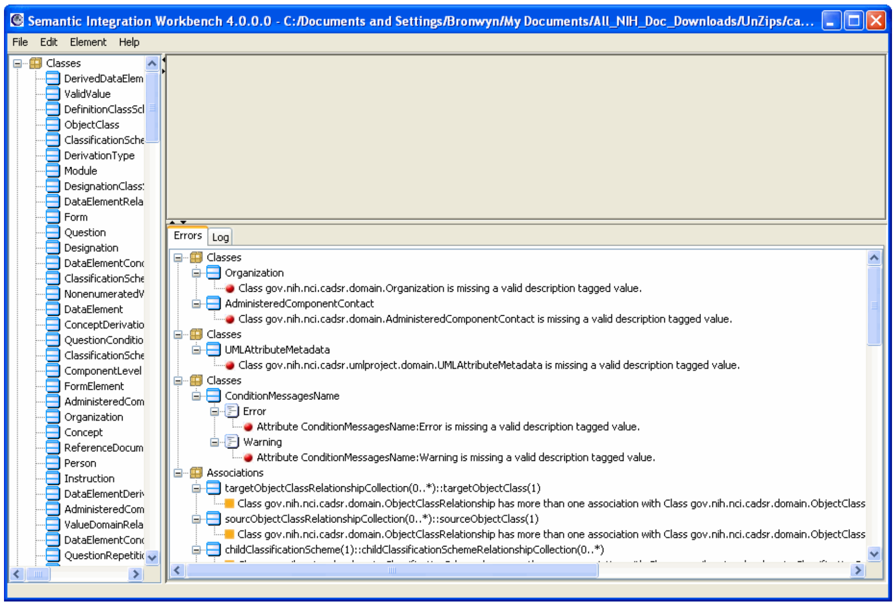
Errors Tab

One valuable feature of the SIW is the ability to quickly isolate and identify errors in the source files, primarily where there is missing or inaccurate information. When the SIW encounters issues while processing a file, those items are flagged as either errors or warnings, and are then displayed in the Errors tab at the bottom of the SIW viewer. The Errors tab provides a description of the error and identifies the source of the error.

Errors are presented using a tree-structure in order to make it easier to locate the source of the error in the navigation tree.

- Errors relevant to a class are shown as a child node to that class' node.
- Errors relevant to an class appear with the full path of the attribute (package/class/attribute/error).

Note: The error tree does not appear if the file contains no errors.



```
Classes
  Organization
    Class gov.nih.nci.cadsr.domain.Organization is missing a valid description tagged value.
  AdministeredComponentContact
    Class gov.nih.nci.cadsr.domain.AdministeredComponentContact is missing a valid description tagged value.
Classes
  UMLAttributeMetadata
    Class gov.nih.nci.cadsr.umlproject.domain.UMLAttributeMetadata is missing a valid description tagged value.
Classes
  ConditionMessagesName
    Error
      Attribute ConditionMessagesName:Error is missing a valid description tagged value.
    Warning
      Attribute ConditionMessagesName:Warning is missing a valid description tagged value.
Associations
  targetObjectClassRelationshipCollection(0..*)::targetObjectClass(1)
    Class gov.nih.nci.cadsr.domain.ObjectClassRelationship has more than one association with Class gov.nih.nci.cadsr.domain.ObjectClass
  sourcObjectClassRelationshipCollection(0..*)::sourceObjectClass(1)
    Class gov.nih.nci.cadsr.domain.ObjectClassRelationship has more than one association with Class gov.nih.nci.cadsr.domain.ObjectClass
  childClassificationScheme(1)::childClassificationSchemeRelationshipCollection(0..*)
```

Figure 2.11 SIW Viewer showing an expanded Errors tab

The information in the Errors tab can be exported by right-clicking in anywhere in the Errors tab and selecting **Export Errors**.

The SIW performs checks for four basic types of errors:

- **Concept Errors** - There are several validation rules that are applied to each model. Any violations of these rules appear in the Errors tab with the source of the error and a description of the problem. Concept errors are most often triggered by missing class or attribute information (missing UML description tagged values or missing concept information).
- **Duplicate Errors** - If two Object Classes (OCs) have the same Public Id or Preferred Name a duplicate mapping error will appear. A duplicate mapping error can also occur if two data elements belonging to the same OC have the same Preferred Name.
- **Association Errors** - Associations that are missing names on the end of the association that has an arrowhead will cause an error. For example, you need a target name for source-->destination associations; you need a both a target name and a source name for bi-directional associations (source <-> destination).
- **Value Domain Errors** - If a definition, Vdtype, datatype, Public Id, or Version are either missing or invalid, an error will appear. If a value domain is in the model but is not used by a data element in the model, this will cause a warning message to appear. Finally, if two value domains are mapped to the same concepts this will also cause an error.

Important Note about updating the Errors tab—The Errors tab information is generated at the time you open the file; the SIW does not update the error list while you are editing. If you have made changes want to regenerate the error listing, save the file and exit the SIW. Then re-open the SIW and select the file you saved. This will update the Errors tab information to reflect the status of the updated file.

Log Tab

The Log tab provides detailed parsing and runtime information about the currently open file. Each log event is represented as a line in the log tab. Log events are organized into four categories:

- Errors: These include errors encountered in the SIW, in the model being parsed or both.
- Warnings
- Info: Events that may be helpful to users
- Debug: Information that may be useful to developers.

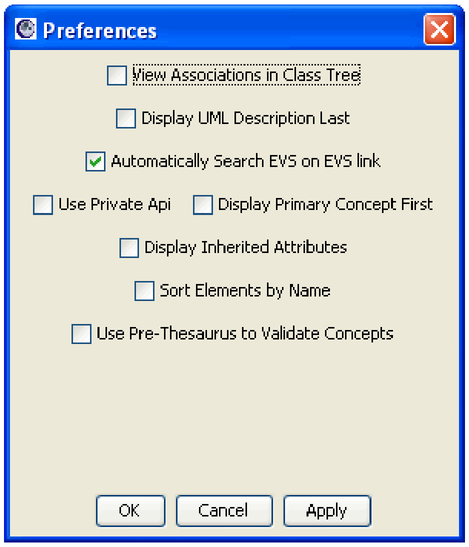
Right-clicking in the Log tab allows you to add or remove any of these categories from the tab. This may be useful if you need to see only certain types of log information.

Setting Viewer Preferences

The SIW viewer allows users to set preferences for using the SIW. Most of the Preferences options available have to do with how the information in the viewer is presented, however a few of the options control the functionality of certain features available through the viewer.

To open the SIW Viewer Preferences dialog box:

- From the main menu of the SIW VIewer, select **Edit > Preferences**.


Preferences

[ ] View Associations in Class Tree

[ ] Display UML Description Last

[✓] Automatically Search EVS on EVS link

[ ] Use Private Api [ ] Display Primary Concept First

[ ] Display Inherited Attributes

[ ] Sort Elements by Name

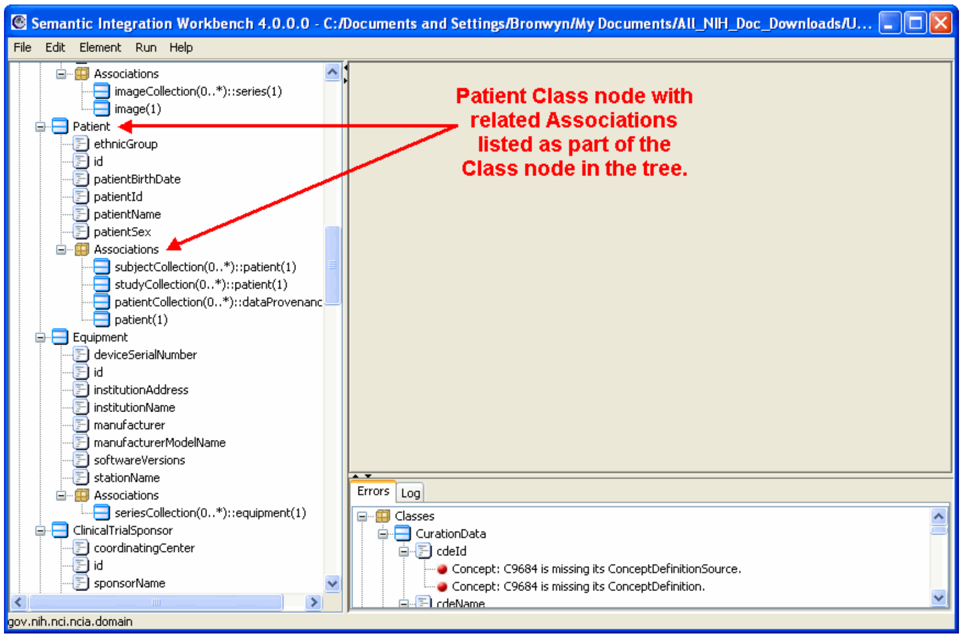
[ ] Use Pre-Thesaurus to Validate Concepts

Figure 2.12 SIW Viewer Preferences dialog box

By default, only the **Automatically Search EVS on EVS Link** option is enabled. All others are disabled by default. Each of the available options is described in more detail in the sections that follow.

View Associations in the Class Tree

By default, model associations are listed all together at the bottom of the navigation tree. Enabling the **View Associations in the Class Tree** option in the SIW Viewer Preferences changes the navigation tree view so that associations appear as children to (nodes beneath) their related classes. This is useful when trying to identify the associations for a particular class.


Semantic Integration Workbench 4.0.0.0 - C:/Documents and Settings/Bronwyn/My Documents/All_NIH_Doc_Downloads/U...
File
Edit
Element
Run
Help
Associations
imageCollection(0..*)::series(1)
image(1)
Patient
ethnicGroup
id
patientBirthDate
patientId
patientName
patientSex
Associations
subjectCollection(0..*)::patient(1)
studyCollection(0..*)::patient(1)
patientCollection(0..*)::dataProvenanc
patient(1)
Equipment
deviceSerialNumber
id
institutionAddress
institutionName
manufacturer
manufacturerModelName
softwareVersions
stationName
Associations
seriesCollection(0..*)::equipment(1)
ClinicalTrialSponsor
coordinatingCenter
id
sponsorName
gov.nih.nci.ncia.domain
Patient Class node with related Associations listed as part of the Class node in the tree.
Errors
Log
Classes
CurationData
cdeId
cdeName

Concept: C9684 is missing its ConceptDefinitionSource.
Concept: C9684 is missing its ConceptDefinition.

Figure 2.13 SIW Viewer with Associations listed under the Class nodes in the Navigation Tree

Note: The View Associations in the Class Tree option does not appear in the Curate XMI mode of the SIW because EVS curators do not work with association elements, and therefore associations do not appear in the SIW viewer in this mode.

Display UML Description Last

As noted in the *Detail View* section of *Using the SIW Viewer* on page 29, by default the UML descriptions for each element are listed first in the detail view, followed by the concept and value domain information where present.

Enabling the option **Display UML Description Last** in the viewer Preferences changes the display so that the UML description appears at the bottom of the detail view instead of the top.

Automatically Search EVS on EVS Link

The **Automatically Search EVS on EVS Link** option is enabled by default. This means that if there is text in the Preferred Concept Name field when you click **Search EVS**, the SIW automatically searches EVS for a synonym of that term and returns the results if any are found.

When this preference is disabled, the search dialog box appears, but no search is run without manually entering search criteria and clicking Search.

Note: If you disable this option and then click **Search EVS**, the previous search information may appear in the search dialog box. This information is NOT cleared between searches. Replace the Search text box text and click Search to search EVS.

Use Private API

Enabling the **Use Private API** Preferences option simply ensures that the **Use Private API** option on the SIW Welcome screen is always checked. This is useful if you typically run the SIW while connected to the NCI LAN (either through VPN or from within the NCI network). It simply removes the need for you to check the checkbox on the Welcome screen.

Display Primary Concept First

As noted in the *Detail View* section of *Using the SIW Viewer* on page 29, by default the Primary Concept information for each element is listed below any Qualifier Concepts mapped to the item, which mimics the semantic order for the name of the element.

Enabling the option **Display Primary Concept First** in the viewer Preferences changes the display so that the Primary Concept appears above the mapped Qualifier Concepts. In addition, if there are multiple Qualifier Concepts for the item, they are now listed in numerical rather than semantic order.

The UML description still appears at the top of the Detail view for each element unless you have enabled the **Display UML Description Last** option (see *Display UML Description Last* above).

Display Inherited Attributes

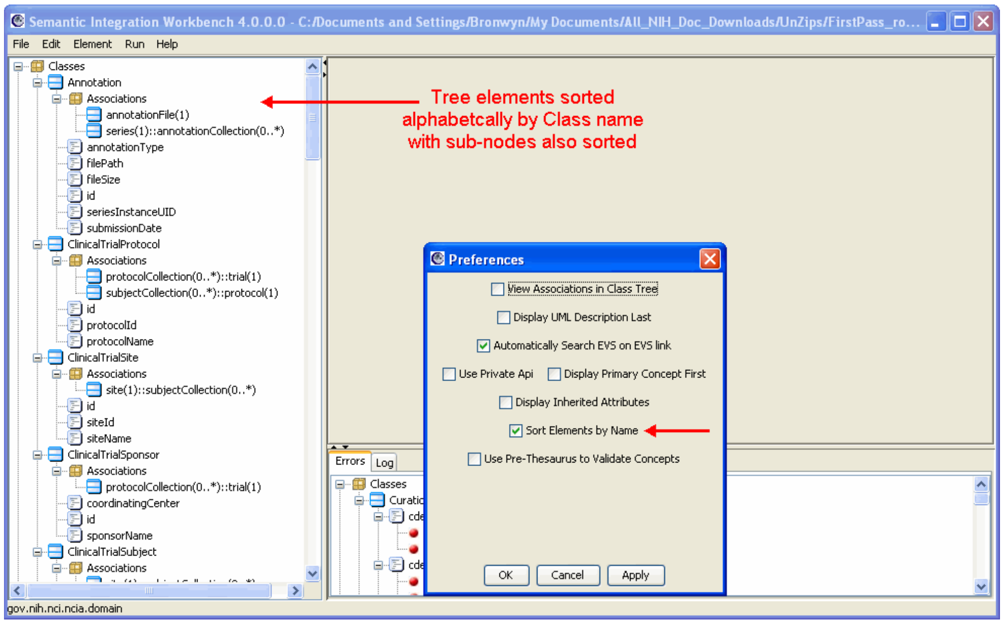
Enabling the **Display Inherited Attributes** changes the navigation tree to show the inherited attributes for any class that inherits attributes from a super class. The attributes appear in an **Inherited Attributes** node below the class.

Enabling this option is useful for if you know you have inherited attributes for class items, in order to call attention to them.

Sort Element by Name

As noted in the *Navigation Tree* section of *Using the SIW Viewer* on page 28, by default the elements in the navigation tree appear in the order in which they exist in the UML model (as indicated in the exported XMI file).

Enabling the option **Sort Element by Name** in the viewer Preferences changes the display so that the elements in the navigation tree are sorted alphabetically. Each class appears in alphabetical order (by name), with each of the elements within the class also listed alphabetically.



Tree elements sorted alphabetcally by Class name with sub-nodes also sorted

Preferences

- [ ] View Associations in Class Tree
- [ ] Display UML Description Last
- [x] Automatically Search EVS on EVS link
- [ ] Use Private Api
- [ ] Display Primary Concept First
- [ ] Display Inherited Attributes
- [x] Sort Elements by Name
- [ ] Use Pre-Thesaurus to Validate Concepts

Deselecting this option resets the navigation tree to again display the classes and their associated elements to their order in the XMI file.

Use Pre-Thesaurus to Validate Concepts

By default, if you select **Validate Concepts** from the Run menu (available in either the Curate XMI File mode or Review Annotated XMI File mode), the SIW validates the concept information in the XMI file against the information in the NCI Production Thesaurus. Enabling the **Use Pre-Thesaurus to Validate Concepts** option causes the SIW to use the Pre-Production NCI Thesaurus to validate concepts instead.

The Pre-Thesaurus is pre-release version of the next release of the NCI Thesaurus, publicly accessible through the DTS server. As is implied, the Pre-Production Thesaurus is more recent than the Production version and as such contains newer concepts. Changing this option in your Viewer Preferences may be useful if you are working with an XMI file that you know contains mapping to newer EVS concepts.

Saving Changes to a File

Clicking **Apply** after making changes to an element is not the same as saving the XMI file. Apply simply applies your changes to the currently open session. You must still save the file using the **File > Save** or **File > Save As** commands to create an updated XMI file containing the applied changes.

Use **File > Save** to overwrite the original file you opened in the SIW viewer. Use **File > Save As** to specify a different name or location for the file.

The Status Bar at the bottom of the SIW viewer window informs you whether or not the file was successfully saved.

XMI files that were created using Enterprise Architect are always saved using an ".xmi" file extension. XMI files created using ArgoUML are always saved using a ".uml" file extension.

Curated and verified files should be saved with the term "annotated" appended to the front of the filename. For example, if the original file name was `FirstPass_myModel.xmi` then you would save the file as `Annotated_FirstPass_myModel.xmi`.

Reviewed and verified files should be saved with the term "approved" appended to the front of the filename. For example, if the original file name was `Annotated_FirstPass_myModel.xmi`, then you would save the reviewed file as `Approved_Annotated_FirstPass_myModel.xmi`.

Updating UML Model Definitions with SIW-Generated Changes

One problem frequently encountered in the semantic integration process is the discovery that the element description tags from the UML model are not present, or that they need to be updated. Prior to the SIW, there was no way to conveniently get these changes into the XMI file for loading to caDSR. Since these tags are mandatory, these tags must be present and should be as accurate as possible.

With the SIW and careful planning, you can change or update the element description tags in the UML model, export them to XMI and go back through the SIW process, while preserving the EVS concept curation before loading the model into caDSR.

In summary, the workflow is as follows:

1. The new UML documentation and description tagged values have to be added to the XMI file using your UML modeling program (EA or ArgoUML). In order to get this information back into the model, import the XMI file you have been working with in SIW.
2. Add the needed information to the model, and then the export the model again.
3. Open the newly exported file in **Review Annotated XMI File** mode. Notice that the SIW contains the new UML description information and still maintains all of the existing EVS concept and caDSR CDE mapping.

As stated in *UML Element Descriptions and Annotations* on page 46, the SIW now recognizes the CADSR_Description tag for use as element description tagged values. You can have up to eight of these tags per element if needed (EA imposes a limit of 255 characters per tag). When re-importing the XMI file back into your UML modeling software, the elements in the model will be annotated with these tagged values.

CHAPTER
3

SEMANTICALLY INTEGRATING YOUR UML MODEL

This chapter provides detailed information on how to use the SIW to semantically integrate your UML Model, making it compatible with other registered systems. The SIW contains five steps or options, and which of those you run depends on the level of caBIG compatibility you wish to achieve and whether you are starting with a new model or you are working with a previously loaded version of your model.

Regardless, the information in this chapter will allow you to use the SIW to create a fully annotated XMI representation of your UML model that can subsequently be loaded into the caDSR (using the UML Loader). The final annotated XMI file can then be used as input to the next version of your system model.

Topics in this chapter include:

Reviewing an Unannotated XMI File

The first option listed on the SIW Welcome screen is the **Review Unannotated XMI File (Model Owner)** option. As indicated, this step is performed by the owner of the UML model.

Reviewing an unannotated XMI file is the process of reviewing an XMI file that has been exported from your UML modeling program (either EA or ArgoUML).

The purpose of viewing your unannotated XMI file through the SIW is so that the program can flag any problems it finds in the model. This allows you to easily review and fix those issues before continuing, and to confirm that the XMI file you may be using for the caCORE SDK is valid for use with the SIW. Typically the errors reported in this mode are due to missing tagged values (e.g., UML description tags).

Input To Step: The original UML model in XMI format, exported from either EA (saved with an .xmi file extension) or ArgoUML (saved with a .uml file extension). The filename can be anything you like, as long as you can browse to and select it.

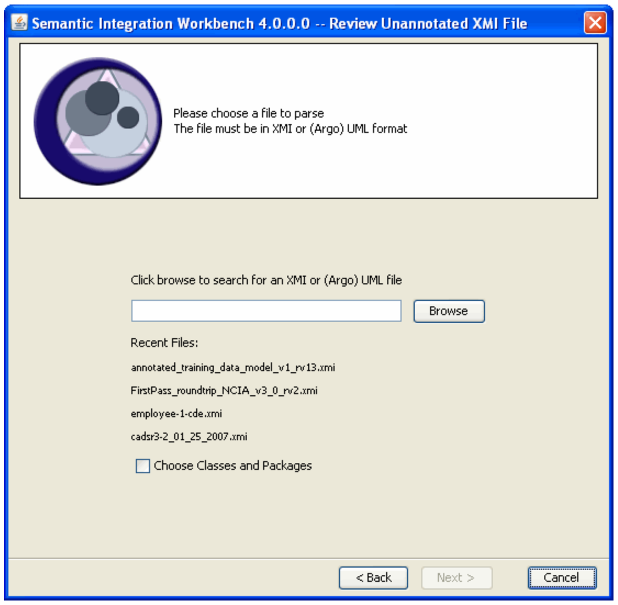
Example: `myModel.xmi` for an EA export; `myModel.uml` for an ArgoUML export.

Output From Step: If you make changes to the file during this step, you can save the file using the **File > Save** or **File > Save As** command. As is typical, using **File > Save** will overwrite the existing file. However, you may want to use **File > Save As** in order to identify that this file differs from the actual UML model output.

While you can save the file using any name you like, we recommend using a naming convention consistent with the naming convention already established for prior releases (if applicable), or a naming convention that will carry to future versions.

To Review an Unannotated XMI file:

1. Be sure you have a valid UML model export file to review. For information on exporting your UML model, see *Generating the XMI file for the SIW* on page 19.
2. Open the SIW URL: http:\\cadsrsiw.nci.nih.gov.
3. From the SIW Welcome screen, select option **1. Review Unannotated XMI File (Model Owner)** and click **Next**. The file selection screen appears.


Semantic Integration Workbench 4.0.0.0 -- Review Unannotated XMI File
Browse
< Back
Next >
Cancel

Please choose a file to parse
The file must be in XMI or (Argo) UML format

Click browse to search for an XMI or (Argo) UML file

Recent Files:

annotated_training_data_model_v1_rv13.xmi
FirstPass_roundtrip_NCIA_v3_0_rv2.xmi
employee-1-cde.xmi
cadsr3-2_01_25_2007.xmi

☐ Choose Classes and Packages

Figure 3.1 Select a file for SIW to parse and display for review

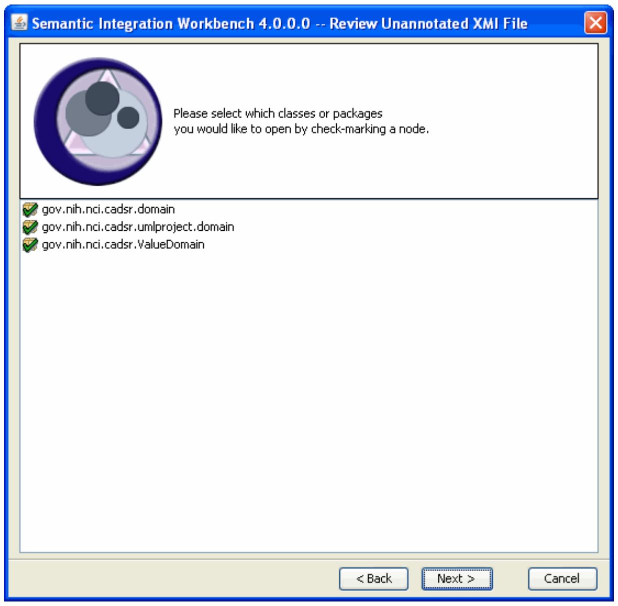
If you have used this mode of the SIW before, the last five files used for this option appear in a Recent Files list located below the filename text box.

4. Identify the file you want to review. You have the following options:
   - If the file you want to review appears in the **Recent Files** list, click on the file name to populate the filename field with the path/filename.
   - **Type** the full path and filename into the text box.
   - Click **Browse** to browse for and select the file exported from EA or ArgoUML.

   Note: The Browse function defaults to selecting (showing) only files with an .xmi file extension. If you are using a .uml file, select **UML Files** from the **Files of Type** drop-down list at the bottom of the Open dialog box.

5. If you want to review only portions of the file, enable the **Choose Classes and Packages** checkbox. Otherwise the SIW will parse the entire file for review.

6. Click **Next**. If you enabled the **Choose Classes and Packages** checkbox, the package selection screen appears, listing all of the packages in the file.


Semantic Integration Workbench 4.0.0.0 -- Review Unannotated XMI File
< Back
Next >
Cancel

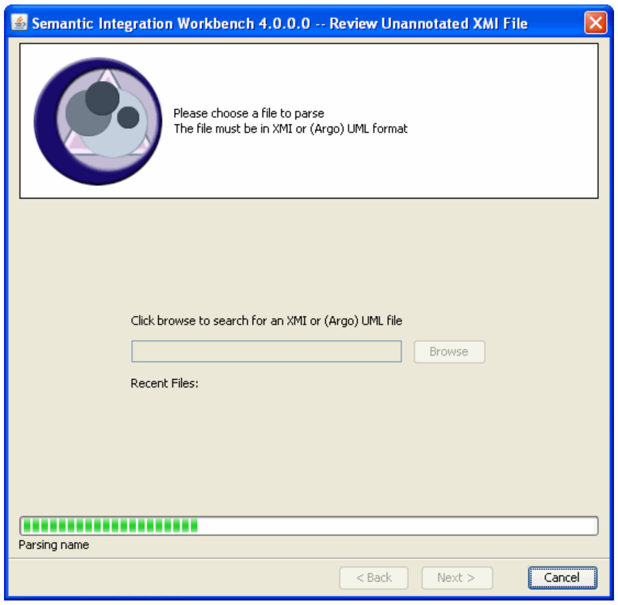
Please select which classes or packages
you would like to open by check-marking a node.

gov.nih.nci.cadsr.domain
gov.nih.nci.cadsr.umlproject.domain
gov.nih.nci.cadsr.ValueDomain

Figure 3.2 Package selection screen - lists the packages resident in the file

7. All packages listed are checked by default. Click a **checkmark** to open a list of the items contained within the package, and then click to clear the checkmarks next to any items you do not want the SIW to parse for review. Then click **Next**.

The SIW displays the file selection screen with a progress bar at the bottom showing the current action and overall progress. How long this process will take is determined by the size of the XMI file or packages you are using as well as the number of errors or warnings the SIW must generate.


Semantic Integration Workbench 4.0.0.0 -- Review Unannotated XMI File
Browse
< Back
Next >
Cancel

Please choose a file to parse
The file must be in XMI or (Argo) UML format

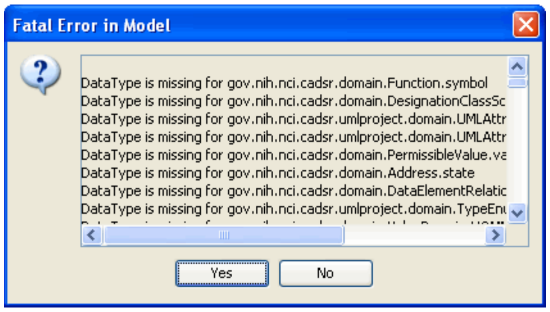
Click browse to search for an XMI or (Argo) UML file

Recent Files:

Parsing name

Figure 3.3 Screen and progress bar showing SIW processing of XMI file

When the SIW is finished, you may receive a Fatal Error in Model dialog box. This lets you know that there are errors in the file that must be fixed before further semantic integration can take place.



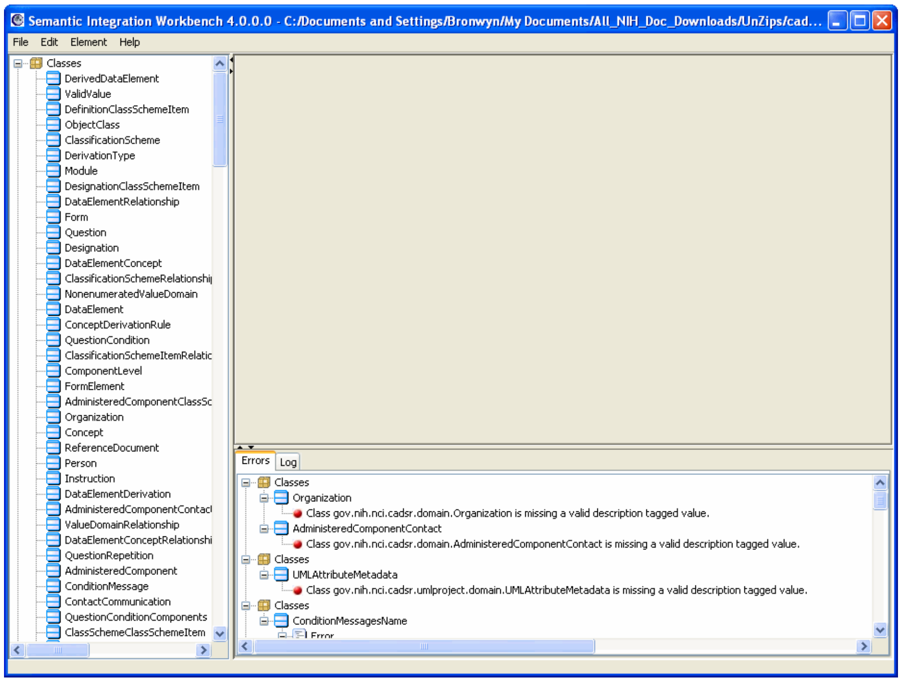
```
DataType is missing for gov.nih.nci.cadsr.domain.Function.symbol
DataType is missing for gov.nih.nci.cadsr.domain.DesignationClassSc
DataType is missing for gov.nih.nci.cadsr.umlproject.domain.UMLAttr
DataType is missing for gov.nih.nci.cadsr.umlproject.domain.UMLAttr
DataType is missing for gov.nih.nci.cadsr.domain.PermissibleValue.va
DataType is missing for gov.nih.nci.cadsr.domain.Address.state
DataType is missing for gov.nih.nci.cadsr.domain.DataElementRelatic
DataType is missing for gov.nih.nci.cadsr.umlproject.domain.TypeEnu
```

Figure 3.4 Fatal Error dialog box identifies the errors found in the XMI file

If the Fatal Error in Model dialog box appears, you have two options:

- **Click No** - This exits the SIW completely. If you know what the problems are, you can fix the errors in your model, re-export and attempt to review the unannotated file again.
- **Click Yes** - This opens the SIW viewer, with the errors and warnings appearing in the Errors tab.

If your file contains no errors, the SIW viewer appears automatically when parsing is finished.



```
Classes
  Organization
    Class gov.nih.nci.cadsr.domain.Organization is missing a valid description tagged value.
  AdministeredComponentContact
    Class gov.nih.nci.cadsr.domain.AdministeredComponentContact is missing a valid description tagged value.
Classes
  UMLAttributeMetadata
    Class gov.nih.nci.cadsr.umlproject.domain.UMLAttributeMetadata is missing a valid description tagged value.
Classes
  ConditionMessagesName
    Error
```

Figure 3.5 SIW Viewer - showing an unannotated XMI file with errors

Use the SIW viewer to review the errors encountered in the exported file, and to determine what changes need to be made in your UML model. Once you have made those changes, you can re-export the XMI file and open it again in the SIW to see if further changes need to be made before continuing.

See *Using the SIW Viewer* on page 27 for more information on how to navigate the SIW viewer to view the various components of your XMI file.

Annotation Basics and Viewing an Unannotated File

The point of viewing an unannotated file in the SIW is to see what items need to be changed in the model (and a new file exported) before you can continue using SIW to annotate the file. This mode of the SIW allows you to be sure that the information resident in the file is clear and accurate and that you have addressed the errors reported by the SIW. This may mean several iterations of file review along with changes to the UML model and re-export to review the new file.

Once the unannotated file is complete, it can move forward through the semantic integration process, providing annotations for the model elements.

The purpose of annotating an XMI file is to map the elements in the file to concept information from EVS. This mapping can only be done if the names of the elements are accurate and the element descriptions provided in the model are sufficiently clear as to the meaning, purpose, or intent of the element.

The following sections provide basic information regarding the relationship between the elements in your UML model and the concept information to which those items will ultimately be mapped.

UML Element Descriptions and Annotations

Each element in the model must have a UML element description. These descriptions are provided through the use of tagged values in the model. The purpose of these descriptions is to provide the EVS concept curators with some understanding as to the model owner's intended meaning of an attribute or class. This allows the curators to more accurately map each element to the appropriate EVS concept(s).

Historically these tagged values were defined as "documentation" tags for classes and "description" tags for attributes. However a new tag has been defined for the SIW called CADSR_Description and can be used as a tagged value for either classes or attributes.

If the UML model contains CADSR_Description tags, that information appears in the SIW as the UML element description. This allows model owners to provide caDSR-specific definitions for each model, which are then pulled into the SIW (and ultimately the caDSR during registration). If the model does not contain these tags, the legacy documentation and description tags are still used.

If there are multiple description tags for an element, the SIW concatenates these tags to display a single UML definition in the SIW. For this concatenation to occur, the format you must use for the CADSR_Description tags is as follows:

- CADSR_Description
- CADSR_Description_2
- CADSR_Description_3

You may add up to eight CADSR_Description tags if necessary (EA has a limit of 255 characters per tag). The UML definition field can include up to 2000 characters.

These descriptions, along with the naming convention used for the elements in the model, provide the information necessary to properly annotate the XMI file.

For a file to be considered “annotated,” each UML element, classes as well as attributes, must be mapped to at least one EVS concept. A concept is made up of the fields described in *Table 3.1*.

Concept Fields	***Example***
Code	`C16612`
Name	`Gene`
Definition	`The physical and functional unit of ...`
Definition Source	`NCI-GLOSS`

Table 3.1 UML element concepts

The process of mapping to EVS concepts is done in two ways:

- Automatically through the Semantic Connector, which uses the name of the element to determine what concepts are appropriate for mapping (applying camel case logic to recognize segments of an element name);
- Manually through the curation process, which is done by the EVS curation team using the element descriptions as a guide to the intended definition, intent, or meaning of each element, mapping the concepts appropriately.

Keep this information in mind as you review your unannotated XMI file. The more accurate and complete the information in the file is, the smoother the annotation process will be.

Association Annotations

Associations are the relationships between items in the UML model. The relationship information for an association can be viewed in the SIW, although it cannot be changed. However, each of the items involved in the relationship (Role, Source, and Target) can have concept information associated with them.

Like all concept annotations, associations elements can have a primary concept with qualifier concepts (as applicable). However, because these concepts are linked to association elements, the tagged values created for each item differ from those created for classes and attributes.

You should consider annotating your associations in order to provide semantic clarity for the associations. In addition, the SIW will return a warning if there are more than two associations between two given classes.

For reference, the following tagged values are used for annotating associations:

1. **To annotate the role:**
   - AssociationRoleConceptCode
   - AssociationRoleConceptPreferredName
   - AssociationRoleConceptDefinition[_n]
   - AssociationRoleConceptDefinitionSource
   - AssociationRoleQualifierConceptCodeN

- AssociationRoleQualifierConceptPreferredNameN
- AssociationRoleQualifierConceptDefinitionN[_n}
- AssociationRoleQualifierConceptDefinitionSourceN

2. **To annotate the source:**
   - AssociationSourceConceptCode
   - AssociationSourceConceptPreferredName
   - AssociationSourceConceptDefinition[_n]
   - AssociationSourceConceptDefinitionSource
   - AssociationSourceQualifierConceptCodeN
   - AssociationSourceQualifierConceptPreferredNameN
   - AssociationSourceQualifierConceptDefinitionN[_n}
   - AssociationSourceQualifierConceptDefinitionSourceN
3. **To annotate the target:**
   - AssociationTargetConceptCode
   - AssociationTargetConceptPreferredName
   - AssociationTargetConceptDefinition[_n]
   - AssociationTargetConceptDefinitionSource
   - AssociationTargetQualifierConceptCodeN
   - AssociationTargetQualifierConceptPreferredNameN
   - AssociationTargetQualifierConceptDefinitionN[_n}
   - AssociationTargetQualifierConceptDefinitionSourceN

Using the XMI Roundtrip Mode

The XMI Roundtrip option is performed by the model owner in an attempt to automatically annotate his or her model with existing data from caDSR. Using this step, a model owner can automatically annotate a new version of a model based on existing mappings located in a previously loaded model. The automated matching is based on the new model having used exactly the same names for the classes and attributes as the previous model.

The XMI Roundtrip is an optional step in the semantic integration process but can save time in the following two situations:

1. A prior version of this model was previously loaded and parts of the models have not changed, specifically when no changes have been made to the names of the classes and attributes.
2. The model shares class and attribute names with another, previously loaded model.

Where possible, the XMI Roundtrip task maps new or updated UML attributes to existing caDSR CDEs rather than to EVS concepts.

Input To Step: The original UML model in XMI format, exported from either EA (saved with an .xmi file extension) or ArgoUML (saved with a .uml file extension) and reviewed through the Review Unannotated XMI File step.

Example: `myModel.xmi` for an EA export; `myModel.uml` for an ArgoUML export.

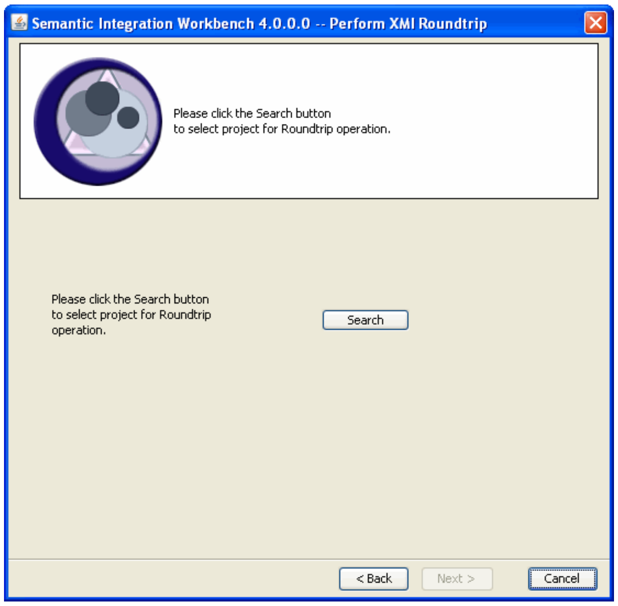
Output From Step: A partially annotated XMI file with the term “Roundtrip” appended to the beginning of the filename (`roundtrip_${filename}.xmi`. or `roundtrip_${filename}.uml`).

The file output from this step differs from a file output from the Semantic Connector in that the Roundtrip XMI file is annotated with caDSR public IDs rather than EVS concepts.

After performing this step, the model owner should review the automated mapping using the **Review Annotated XMI File** option. See *Reviewing an Annotated XMI File* on page 74.

To Perform XMI Roundtrip:

1. Open the SIW URL: http:\\cadsrsiw.nci.nih.gov.
2. From the SIW Welcome screen, select option **2. Perform XMI Roundtrip (Model Owner)** and click **Next**. The Project Search screen appears.


Semantic Integration Workbench 4.0.0.0 -- Perform XMI Roundtrip

Please click the Search button
to select project for Roundtrip operation.

Please click the Search button
to select project for Roundtrip
operation.

Figure 3.6 Search for an existing Project to use for Roundtrip

3. In the project search screen, click **Search**. This opens a Search for Classification Schemes window (*Figure 3.7*) that allows you to search caDSR for an existing project and version to use for the Roundtrip processing.

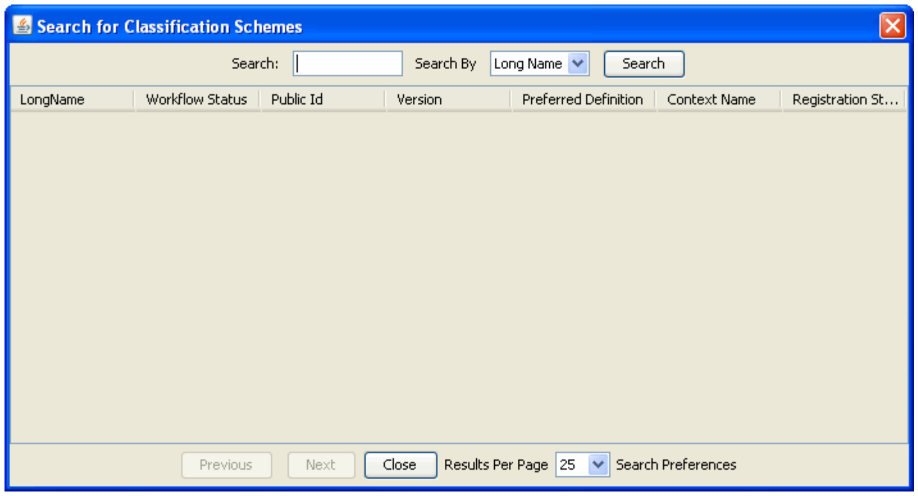
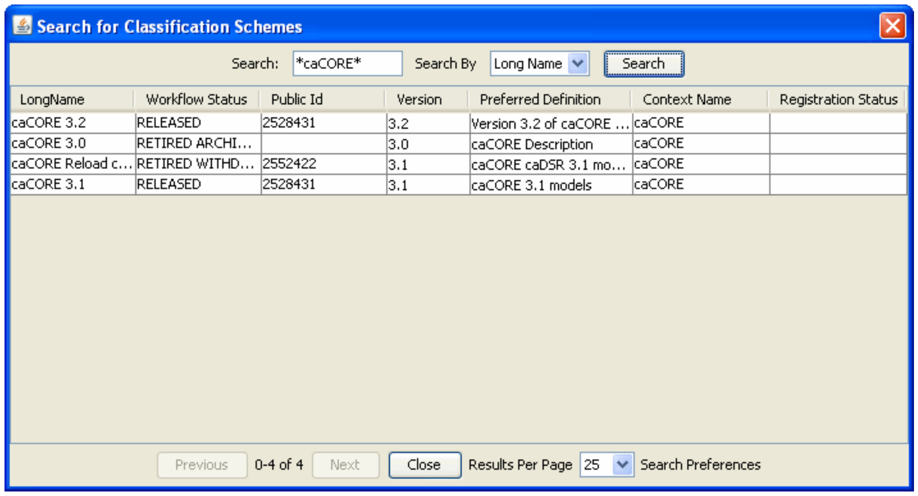


Figure 3.7 Search for Classification Schemes (project) to use for Roundtrip processing

4. Enter a classification scheme name in the Search text box and click **Search**. You may use the asterisk (*) as a wildcard along with a portion of a name, to return a list of projects to choose from.



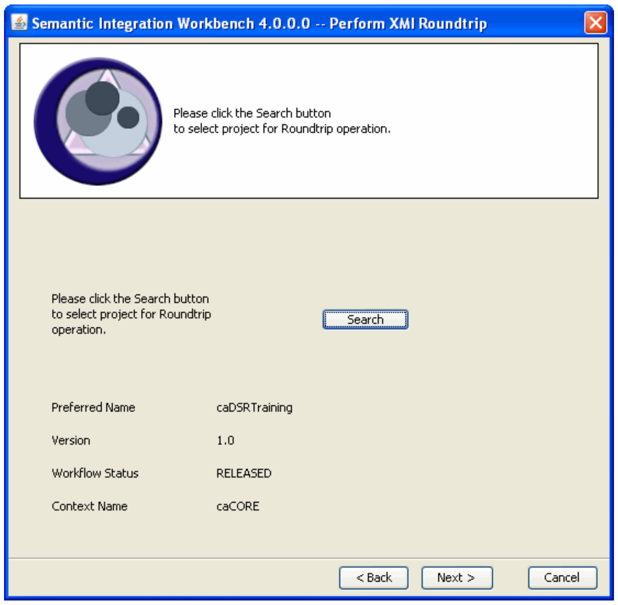
Search: *caCORE*

LongName	Workflow Status	Public Id	Version	Preferred Definition	Context Name	Registration Status
caCORE 3.2	RELEASED	2528431	3.2	Version 3.2 of caCORE ...	caCORE	
caCORE 3.0	RETIRED ARCHI...		3.0	caCORE Description	caCORE	
caCORE Reload c...	RETIRED WITHD...	2552422	3.1	caCORE caDSR 3.1 mo...	caCORE	
caCORE 3.1	RELEASED	2528431	3.1	caCORE 3.1 models	caCORE	

0-4 of 4

Figure 3.8 Search for Classification Schemes - result set

5. From the result set, find the project *and* version of the project you want to use for processing your new file through Roundtrip, and double-click on that entry. The select project window returns, showing the details of the selected project.


Semantic Integration Workbench 4.0.0.0 -- Perform XMI Roundtrip
Search
< Back
Next >
Cancel

Please click the Search button to select project for Roundtrip operation.

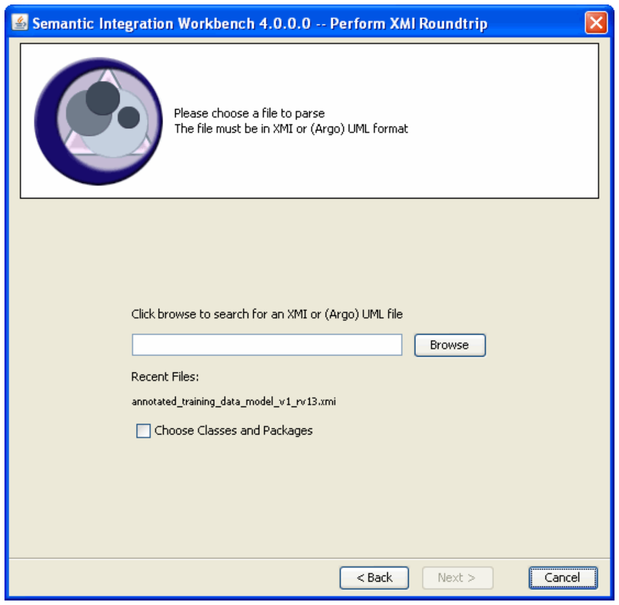
Please click the Search button to select project for Roundtrip operation.

Preferred Name	caDSRTraining
Version	1.0
Workflow Status	RELEASED
Context Name	caCORE

Figure 3.9 Project selection screen showing selected Roundtrip project details

6. Click **Next**.
7. In the Select file to parse window, identify the file you want to process. You have the following options:
   - If the file you want to review appears in the **Recent Files** list, click on the file name to populate the filename field with the path/filename;
   - **Type** the full path and filename into the text box;
   - Click **Browse** to browse for and select the file exported from EA or ArgoUML.

Note: The Browse function defaults to selecting (showing) only files with an .xmi file extension. If you are using a .uml file, select **UML Files** from the **Files of Type** drop-down list at the bottom of the Open dialog box.


Semantic Integration Workbench 4.0.0.0 -- Perform XMI Roundtrip
Browse
< Back
Next >
Cancel

Please choose a file to parse
The file must be in XMI or (Argo) UML format

Click browse to search for an XMI or (Argo) UML file

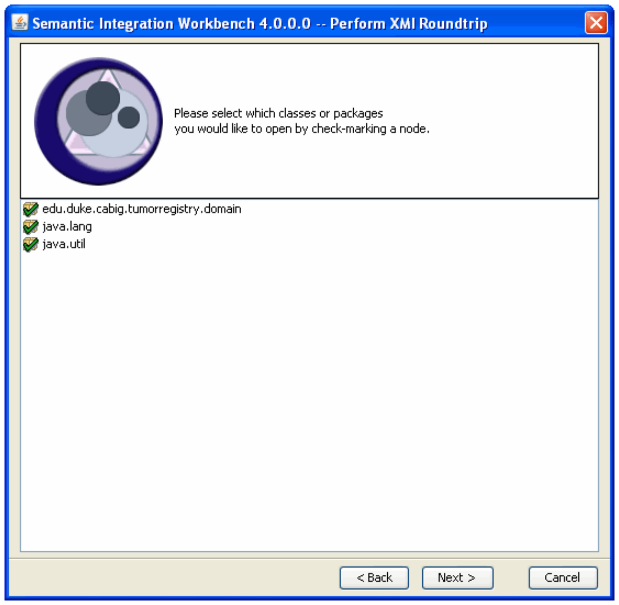
Recent Files:

annotated_training_data_model_v1_rv13.xmi

☐ Choose Classes and Packages

Figure 3.10 Select the file to process through XMI Roundtrip

8. If you want to process only portions of the file, enable the **Choose Classes and Packages** checkbox. Otherwise the SIW will submit the entire file through the Roundtrip process.
9. Click **Next**. If you enabled the **Choose Classes and Packages** checkbox, the package selection screen appears, listing all of the packages in the file. All packages are checked by default.


Semantic Integration Workbench 4.0.0.0 -- Perform XMI Roundtrip
< Back
Next >
Cancel

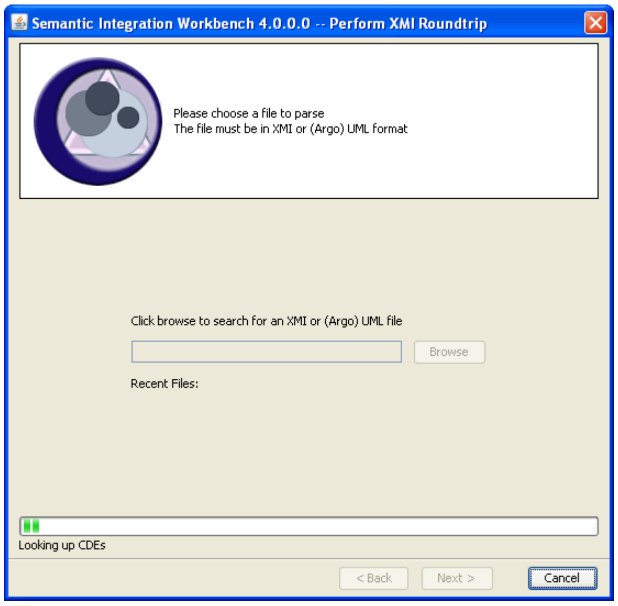
Please select which classes or packages
you would like to open by check-marking a node.

✓ edu.duke.cabig.tumorregistry.domain
✓ java.lang
✓ java.util

Figure 3.11 Package selection screen - lists the packages resident in the file

10. Click a **checkmark** to open a list of all items contained within the package, and then click to remove the checkmarks next to any items you do not want submitted through Roundtrip. Then Click **Next**.

The Roundtrip step begins to process your XMI file against the project you selected.


Semantic Integration Workbench 4.0.0.0 -- Perform XMI Roundtrip
Browse
Recent Files:
< Back
Next >
Cancel

Please choose a file to parse
The file must be in XMI or (Argo) UML format

Click browse to search for an XMI or (Argo) UML file

Looking up CDEs

Figure 3.12 SIW performs the Roundtrip process on the selected file against the selected project

For each UML attribute in the XMI file being processed, the SIW attempts to find a CDE with an Alternate Name in the selected Project/Version that is similar to the fully qualified attribute name in the model. If a match is found, the attribute in the XMI file is annotated with the matching CDE public ID/Version from caDSR. Note that mapping an attribute to a CDE also maps the class to a caDSR OC and the datatype for the attribute to a caDSR value domain.

When the Roundtrip process is complete, the SIW saves a new version of the XMI file into the same directory as the source file, appended with the term “Roundtrip” on the front of the filename. For example, if the processed file’s path/name was `c:\myXMIfiles\Version2\myModel.xmi` the newly saved file will be `c:\myXMIfiles\Version2\Roundtrip_myModel.xmi`.

After performing this step, the model owner should review the automated mapping using the **Review Annotated XMI File** option. After that, if there are UML elements in the file that are not mapped, the model owner may want to process the XMI file through the Semantic Connector to map those elements to EVS concepts.

For more information, see *Running the Semantic Connector*, below, and *Reviewing an Annotated XMI File* on page 74.

Running the Semantic Connector

The Semantic Connector option of the SIW performs an EVS search against each element in the XMI file (or package), and attaches one or more EVS concepts to each element. The search is performed using the element names. If the elements in the file are named properly using the “camel case” convention, the SIW can separate the parts of each element based on the capitalization in the element name, and perform an EVS search on each part. The Semantic Connector then maps the parts of each element to EVS concepts as appropriate.

For example, the XMI file contains an attribute called “initialDiagnosisDate.” Because the element name uses camel case, the Semantic Connector recognizes three parts to the element: initial, diagnosis and date. Each of those terms is then searched against the NCI Thesaurus, and three matching EVS concepts are returned.

Because “date” appears last in the series, the Semantic Connector identifies that as being the main idea of the element, and therefore maps the Primary Concept as “date.” The Qualifier Concept #1 is mapped as “diagnosis” and Qualifier Concept #2 is mapped as “initial.” The element is now annotated with a primary and two qualifier concepts.

When the Semantic Connector step is finished, the SIW automatically creates the Semantic Connector Report and saves it as a new file to the same directory where the source file resides. The model owner should review this file before submitting it to the EVS curators for the Curate XMI File process.

Although the Semantic Connector will add concepts to anything not marked as “reviewed” (even those items already mapped to a CDE), those concepts are ignored during validation and loading, in favor of the mapped CDEs.

Input To Step: The original UML model in XMI format, exported from either EA (saved with an .xmi file extension) or ArgoUML (saved with a .uml file extension) and reviewed through the Review Unannotated XMI File step. Or the input could be an XMI file that has been processed through the XMI Roundtrip step.

Example: `myModel.xmi` for an EA export; `myModel.uml` for an ArgoUML export; or `roundtrip_myModel.xmi` or `roundtrip_myModel.uml` for a Roundtrip processed file.

Output From Step: The Semantic Connector Report, saved as an XMI file with the term “FirstPass” appended to the front of the file name (e.g., `FirstPass_myModel.xmi` or `FirstPass_Roundtrip_myModel.xmi`).

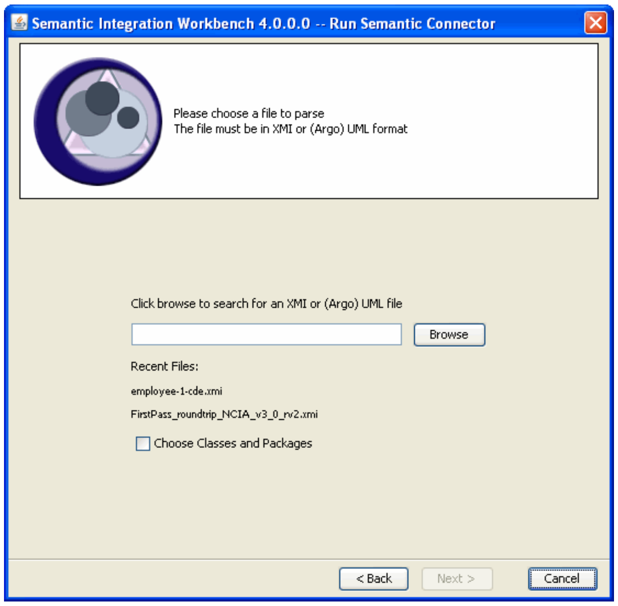
After performing this step, the model owner should review the automated mapping using the **Review Annotated XMI File** option. See *Reviewing an Annotated XMI File* on page 74.

The file output from this step is sent to the EVS curation team as the input for the Curate XMI File step of the SIW.

To run the Semantic Connector:

1. Open the SIW URL: http:\\cadsrsiw.nci.nih.gov.
2. From the SIW Welcome screen, select option **3. Run Semantic Connector (Model Owner)** and click **Next**.
3. In the Select file to parse window, identify the file you want to process. You have the following options:
   - If the file you want to review appears in the **Recent Files** list, click on the file name to populate the filename field with the path/filename.
   - **Type** the full path and filename into the text box.
   - Click **Browse** to browse for and select the file exported from EA or ArgoUML.

   Note: The Browse function defaults to selecting (showing) only files with an .xmi file extension. If you are using a .uml file, select **UML Files** from the **Files of Type** drop-down list at the bottom of the Open dialog box.


Semantic Integration Workbench 4.0.0.0 -- Run Semantic Connector
Browse
< Back
Next >
Cancel

Please choose a file to parse
The file must be in XMI or (Argo) UML format

Click browse to search for an XMI or (Argo) UML file

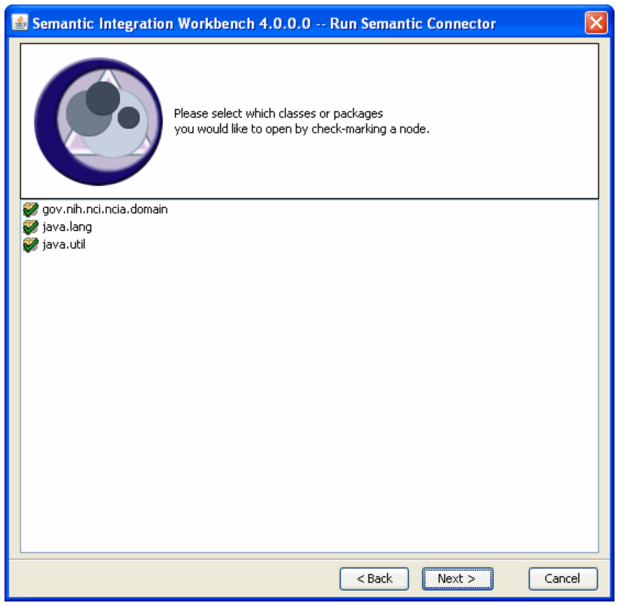
Recent Files:
employee-1-cde.xmi
FirstPass_roundtrip_NCIA_v3_0_rv2.xmi

☐ Choose Classes and Packages

Figure 3.13 Select the file to process through the Semantic Connector

4. If you want to process only portions of the file, enable the **Choose Classes and Packages** checkbox. Otherwise the SIW will submit the entire file through the Semantic Connector.

5. Click **Next**. If you enabled the **Choose Classes and Packages** checkbox, the package selection screen appears, listing all of the packages in the file. All packages are checked by default.


Semantic Integration Workbench 4.0.0.0 -- Run Semantic Connector

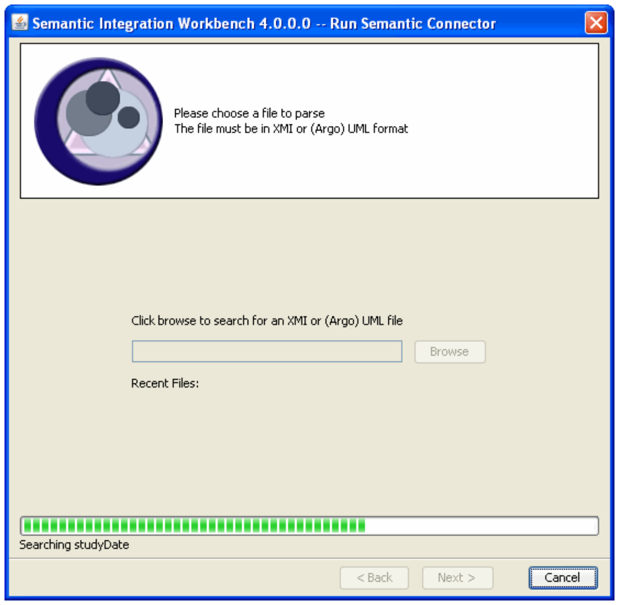
Please select which classes or packages
you would like to open by check-marking a node.

✓ gov.nih.nci.ncia.domain
✓ java.lang
✓ java.util

Figure 3.14 Package selection screen - lists the packages resident in the file

6. Click a **checkmark** to open a list of all items contained within the package, and then click to remove the checkmarks next to any items you do not want to process through the Semantic Connector. Then click **Next**.

The Semantic Connector begins to process the XMI file against the NCI Thesaurus.


Semantic Integration Workbench 4.0.0.0 -- Run Semantic Connector

Please choose a file to parse
The file must be in XMI or (Argo) UML format

Click browse to search for an XMI or (Argo) UML file

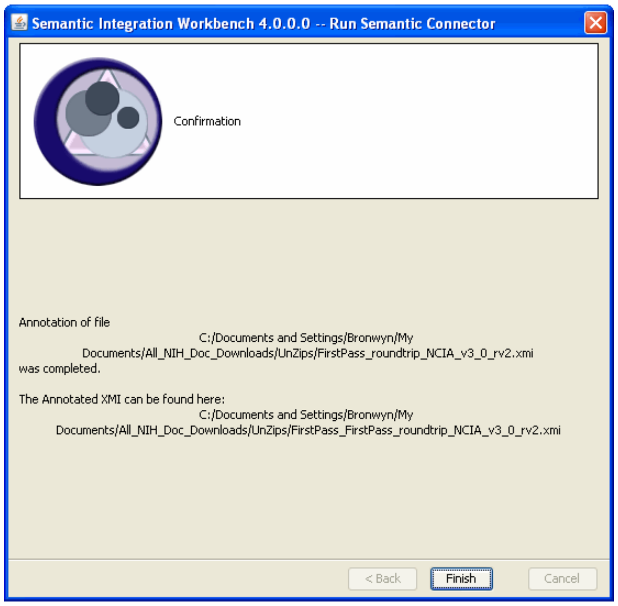
Recent Files:

Searching studyDate

Figure 3.15 SIW performs the Roundtrip process on the selected file

If the process is successful, the Semantic Connector maps the elements to EVS concepts and inserts them into the XMI file. When the Semantic Connector is done, the SIW saves a new version of the XMI file, which is referred to as the Semantic

Connector Report. A final screen appears confirming completion of the process and identifying the location of the newly generated file. Note the location and click **Finish**.


Semantic Integration Workbench 4.0.0.0 -- Run Semantic Connector
Confirmation
< Back
Finish
Cancel

Annotation of file
C:/Documents and Settings/Bronwyn/My Documents/All_NIH_Doc_Downloads/UnZips/FirstPass_roundtrip_NCIA_v3_0_rv2.xmi
was completed.

The Annotated XMI can be found here:
C:/Documents and Settings/Bronwyn/My Documents/All_NIH_Doc_Downloads/UnZips/FirstPass_FirstPass_roundtrip_NCIA_v3_0_rv2.xmi

Figure 3.16 Semantic Connector process complete - displays location of new XMI file

This new file is saved into the same directory as the source file and is appended with the term "FirstPass" on the front of the filename. For example, if the processed file's path/name was `c:\myXMIfiles\myModel.xmi` the newly saved file will be `c:\myXMIfiles\FirstPass_myModel.xmi`. If the example file had been submitted through the Roundtrip process first, the newly saved file would be named `c:\myXMIfiles\FirstPass_Roundtrip_myModel.xmi`. If your source was exported from ArgoUML, the extension on the filename would continue to be `.uml`.

After performing this step, the model owner should review the Semantic Connector Report (new XMI file) using the **Review Annotated XMI File** option. For more information, see *Reviewing an Annotated XMI File* on page 74.

After reviewing the changes, the file should be submitted to the EVS curation team to perform the Curate XMI File step of the SIW process. The EVS curation team reviews the file, removes extraneous mapped concepts, inserts new concepts, and reorders the mapped concepts to match the UML class and attribute entities and the intent identified for the element by the owner (through the element description tags).

Note: If the Semantic Connector process was not successful, an error message may appear, or the process may simply not complete. The most likely cause for this is that the XMI file is not in the expected format. This issue should be corrected before re-running the Semantic Connector and may require re-export of the XMI file from EA.

Curating XMI Files

The **Curate XMI File** step is performed by the EVS concept curation team at the NCICB. During this step, the EVS team adds and removes concepts from the XMI file, and indicates recommended semantic mappings for the UML model. The EVS team will use existing EVS concepts where possible, or create new EVS concepts to reflect new items being introduced by the model.

EVS curators have the authority to change any concept field and to use concepts that are not in the NCI Thesaurus. Model owners, on the other hand, should only change concept information by picking an EVS concept from the Thesaurus or entering one supplied by EVS. New EVS concepts cannot be verified using the **Search EVS** feature, so new concepts should only be used by EVS curators during the Curate XMI File step. However if the EVS curation team supplies the model owner with a new EVS concept to use, they can be added during the Review Annotated XMI File step.

The end result of the curation process is an Annotated XMI file that is returned to the model owner who can then review the annotated file and either accept or change the mappings performed by the curators. Frequently, if the model owner makes changes to the file, they will return it to the EVS curation team for re-review and curation.

Input: The Semantic Connector Report, which is the file that the SIW saved after completing the Run Semantic Connector step. The file should be appended with “FirstPass” at the front of the filename.

Example: `FirstPass_myModel.xmi` or `FirstPass_Roundtrip_myModel.xmi` (if the file was processed through the Roundtrip step before the Semantic Connector).

Output: An curated XMI file. The file should be appended with “annotated” at the front of the filename. For example: `Annotated_FirstPass_myModel.xmi` or `Annotated_FirstPass_Roundtrip_myModel.xmi`.

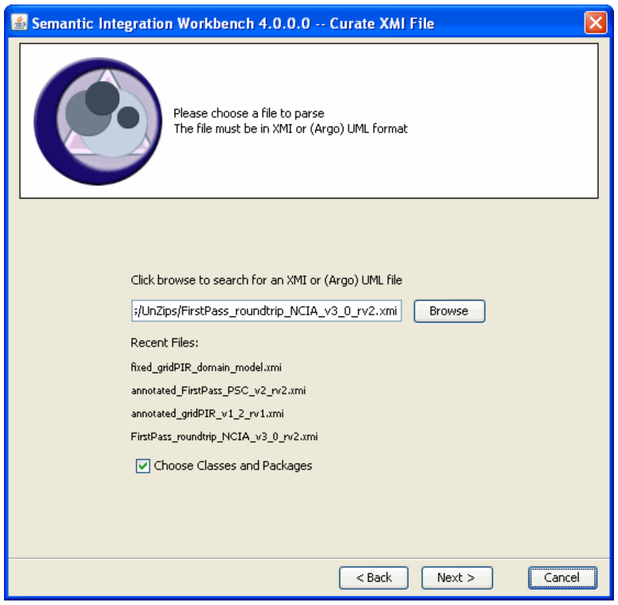
The curated XMI file is the input to the Review Annotated Model step. For more information, see *Reviewing an Annotated XMI File* on page 74.

Note: The procedural information in this section is directed specifically at EVS curators, though model owners are *strongly* encouraged to review this section in order to understand the process of curation. Be advised however, where the text in this section refers to “you” or “the user” performing a task, it is referring to the EVS curator.

To Curate an XMI File:

1. Open the SIW URL: http:\\cadsrsiw.nci.nih.gov.
2. From the SIW Welcome screen, select option **4. Curate XMI File (Vocabulary Reviewer)** and click **Next**.
3. In the Select file to parse window, identify the file you want to process. You have the following options:
    - If the file you want to review appears in the **Recent Files** list, click on the file name to populate the filename field with the path/filename.
    - **Type** the full path and filename into the text box.

- Click **Browse** to browse for and select the file exported from EA or ArgoUML.


Semantic Integration Workbench 4.0.0.0 -- Curate XMI File

Please choose a file to parse
The file must be in XMI or (Argo) UML format

Click browse to search for an XMI or (Argo) UML file

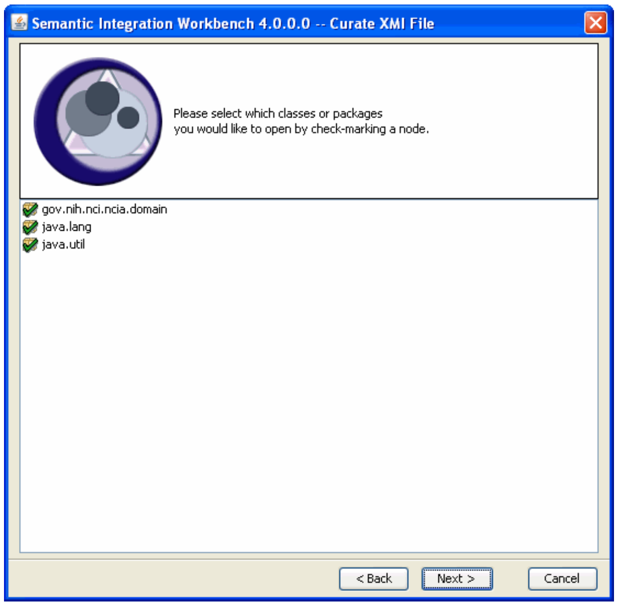
:/UnZips/FirstPass_roundtrip_NCIA_v3_0_rv2.xmi

Recent Files:
fixed_gridPIR_domain_model.xmi
annotated_FirstPass_PSC_v2_rv2.xmi
annotated_gridPIR_v1_2_rv1.xmi
FirstPass_roundtrip_NCIA_v3_0_rv2.xmi

[✓] Choose Classes and Packages

Figure 3.17 Select the XMI file to curate

4. If you want to process only portions of the file, enable the **Choose Classes and Packages** checkbox. Otherwise the SIW will open the entire file for curation.
5. Click **Next**. If you enabled the **Choose Classes and Packages** checkbox, the package selection screen appears, listing all of the packages in the file. All packages are checked by default.


Semantic Integration Workbench 4.0.0.0 -- Curate XMI File

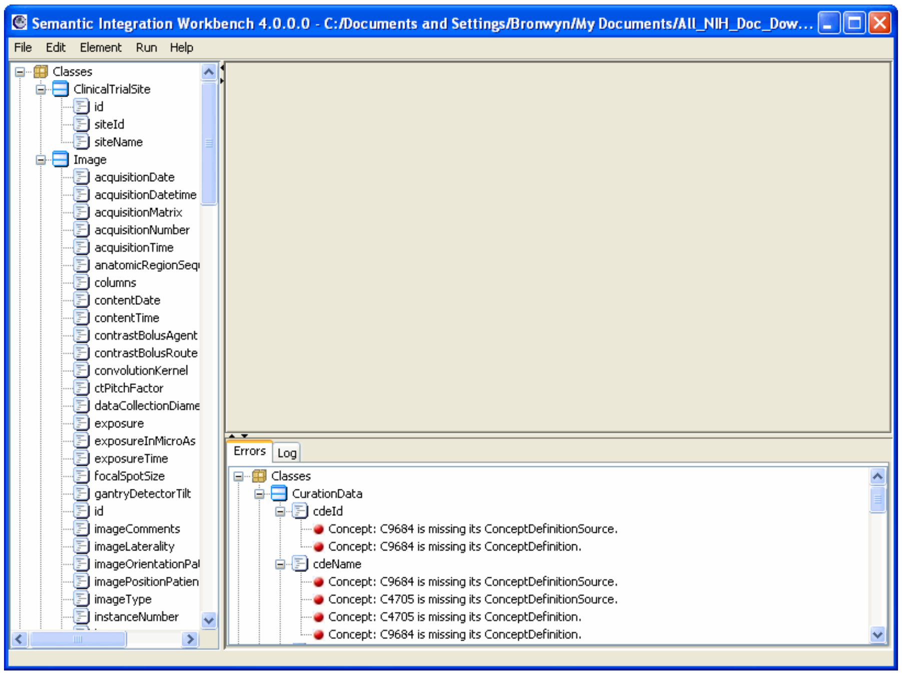
Please select which classes or packages
you would like to open by check-marking a node.

✓ gov.nih.nci.ncia.domain
✓ java.lang
✓ java.util

Figure 3.18 Package selection screen - lists the packages resident in the file

6. Click a **checkmark** to open a list of all items contained within the package, and then click to remove the checkmarks next to any items you do not want to open for curation. Then click **Next**.

When the SIW is finished parsing the file, the SIW viewer appears.


Semantic Integration Workbench 4.0.0.0 - C:/Documents and Settings/Bronwyn/My Documents/All_NIH_Doc_Dow...
File
Edit
Element
Run
Help
Errors
Log

```
Classes
  ClinicalTrialSite
    id
    siteId
    siteName
  Image
    acquisitionDate
    acquisitionDatetime
    acquisitionMatrix
    acquisitionNumber
    acquisitionTime
    anatomicRegionSeq
    columns
    contentDate
    contentTime
    contrastBolusAgent
    contrastBolusRoute
    convolutionKernel
    ctPitchFactor
    dataCollectionDiame
    exposure
    exposureInMicroAs
    exposureTime
    focalSpotSize
    gantryDetectorTilt
    id
    imageComments
    imageLaterality
    imageOrientationPat
    imagePositionPatien
    imageType
    instanceNumber
```

```
Classes
  CurationData
    cdeId
      Concept: C9684 is missing its ConceptDefinitionSource.
      Concept: C9684 is missing its ConceptDefinition.
    cdeName
      Concept: C9684 is missing its ConceptDefinitionSource.
      Concept: C4705 is missing its ConceptDefinitionSource.
      Concept: C4705 is missing its ConceptDefinition.
      Concept: C9684 is missing its ConceptDefinition.
```

Figure 3.19 SIW Viewer - showing an XMI file ready for curation

Any element not already mapped to a caDSR CDE or OC must be mapped to one or more EVS concepts. The Semantic Connector process did some of the concept mapping automatically, however it likely mapped multiple concepts for more common elements (such as “id”) or did not find any matching concepts for some elements. Furthermore, additional concepts may need to be added to an element, some mapped concepts may need to be removed, or the semantic order of the concepts may need to change.

The SIW viewer allows you to add, remove, edit and re-order the EVS concepts for the elements in the XMI file. The sections that follow provide information on how to perform these tasks.

If necessary, see *Using the SIW Viewer* on page 27 for a basic overview on navigating the SIW viewer and on using the **Previous**, **Next**, **Add**, **Apply** and **Search EVS** buttons.

Adding EVS Concepts to an Element in Curate Mode

If an element does not have an EVS concept mapped to it, you must add one. If an element has an EVS concept mapped to it, you may add one or more Qualifier concepts to the existing concept(s).

The first concept added to an element is called the “Primary” concept. Subsequent mapped concepts are called “Qualifier” concepts. Concepts are added and then listed in semantic order, meaning that the Primary concept is always added first and must reflect the basic point of the element (typically the last item in the element name).

For example, an element is named “initalDiagnosisDate.” The item itself reflects a date, while the rest of the information reveals details about the date. In this case, the Primary concept for the element should be mapped to an EVS concept named “date.”

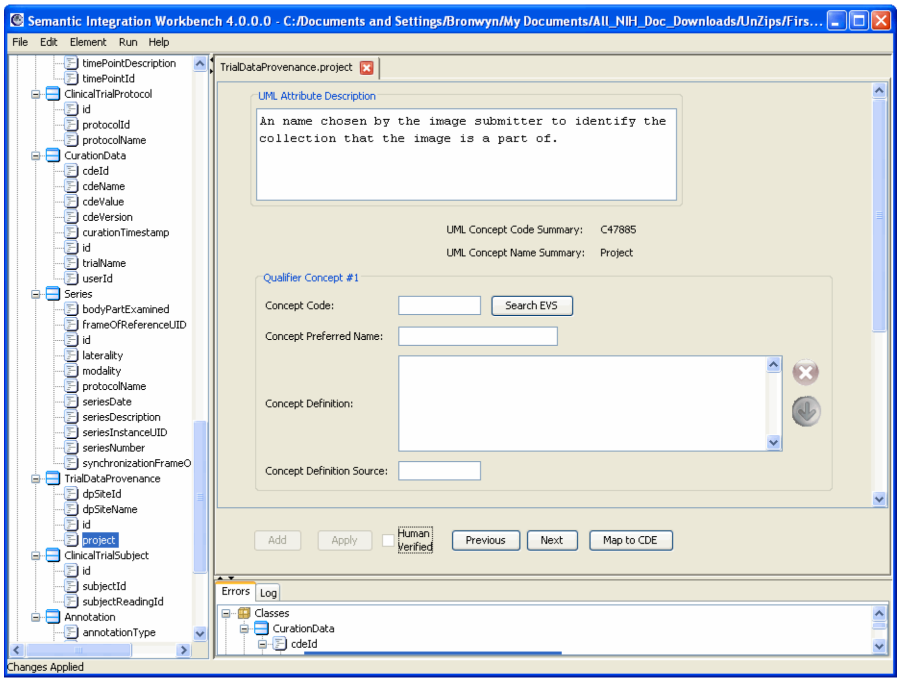
If the element already has a Primary concept mapped, it may require further mapping of Qualifier concepts. In the example above, once the Primary concept is mapped, you would add a Qualifier concept mapped to an EVS concept named “diagnosis” and then add a second Qualifier concept mapped to an EVS concept named “initial.”

Note: The procedures for adding concepts to an element are the same, regardless of whether you are adding Primary or a Qualifier concepts. Furthermore, adding concepts only applies to elements that are not already mapped to caDSR CDEs or OCs.

To add a concept:

1. In the SIW viewer, select an element from the navigation tree.
2. In the detail view, click **Add**. A set of empty concept fields appears, along with a **Search EVS** button.

   Note: If you have changed your viewer Preferences to Display Primary Concept First, the new concept fields will appear below any existing mapped concepts, meaning you may need to scroll down in the detail view to see the new fields. See *Setting Viewer Preferences* on page 36 for more information on setting your viewer preferences.



An name chosen by the image submitter to identify the collection that the image is a part of.

UML Concept Code Summary: C47885

UML Concept Name Summary: Project

Figure 3.20 SIW Viewer in Curate XMI mode, showing new blank concept fields.

3. Enter the appropriate information into all four text fields. You have the following options:

   - Manually type the information into the text boxes,

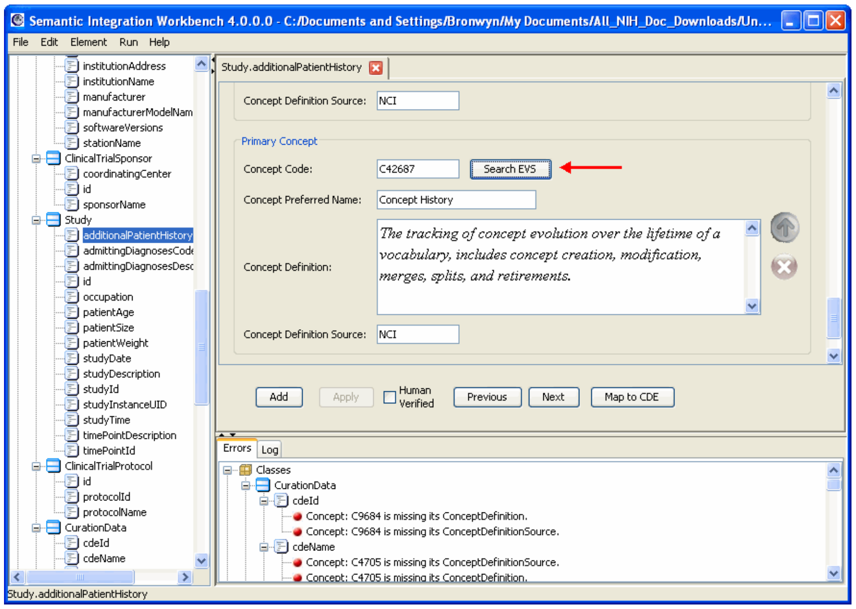
- Click **Search EVS** to perform a search for the appropriate concept to map. **Double-click on a search result item** to populate the concept fields with the needed information. For more details on using the EVS search to populate the concept fields, see *Searching EVS for Concept Information*, below.

4. When the fields are complete, click **Apply**. This applies the new concept to the the element. (You will still have to save the file after all of your changes are made. See *Saving Changes to a File* on page 39.
5. If necessary, after applying your changes, you can click **Add** again and complete the above steps to add another concept to this element.

Searching EVS for Concept Information

Whether adding or editing concept information, the detail view of the SIW viewer provides a **Search EVS** button that allows you to search the NCI Thesaurus for existing concepts and if a match is found, to automatically map the concept to the element.

The **Search EVS** button appears next to each mapped concept and next to the empty concept information fields when you select to Add a new concept.



Concept Definition Source: NCI

Concept Code: C42687

Concept Preferred Name: Concept History

Concept Definition: *The tracking of concept evolution over the lifetime of a vocabulary, includes concept creation, modification, merges, splits, and retirements.*

Concept Definition Source: NCI

```
Classes
  CurationData
    cdeId
      Concept: C9684 is missing its ConceptDefinition.
      Concept: C9684 is missing its ConceptDefinitionSource.
    cdeName
      Concept: C4705 is missing its ConceptDefinitionSource.
      Concept: C4705 is missing its ConceptDefinition.
```

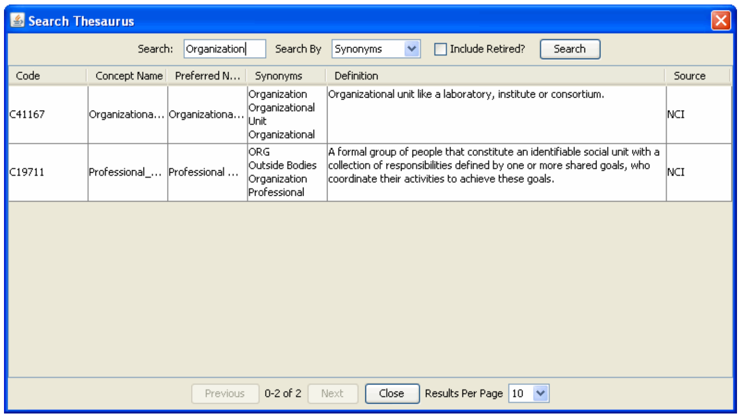
Figure 3.21 The Search EVS button appears next to concept information fields

By default, if the Concept Name field is populated when you click **Search EVS**, the EVS search automatically performs a synonym search of the NCI Thesaurus for the concept name listed. If you have changed your viewer Preferences to not automatically perform an EVS search, then you will need to manually enter search criteria. See *Setting Viewer Preferences* on page 36 for more information.

To search EVS/NCI Thesaurus for concept information:

1. In the SIW viewer, click **Search EVS**, located to the right of the Concept Code field. The Search Thesaurus dialog box appears in one of two states:

- If the Concept Name field was populated when you clicked **Search EVS** (and your Viewer Preferences are set to automatically search EVS), the concept name appears in the Search text box, a search is automatically performed, and the results appear in the Search Thesaurus dialog box.

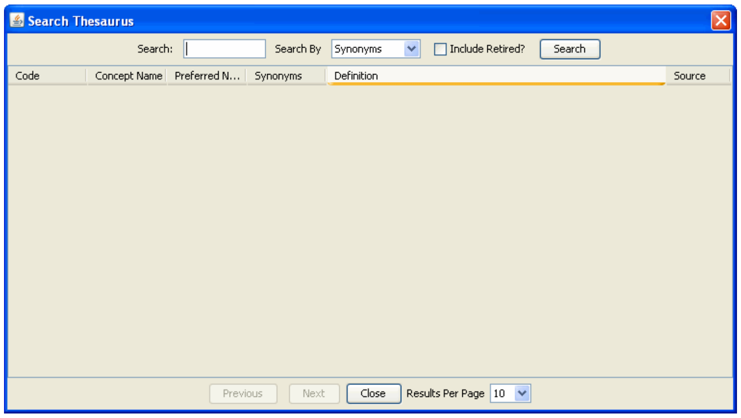


Search: Organization | Search By: Synonyms | Include Retired? ☐ | Results Per Page: 10

Code	Concept Name	Preferred N...	Synonyms	Definition	Source
C41167	Organizationa...	Organizationa...	Organization Organizational Unit Organizational	Organizational unit like a laboratory, institute or consortium.	NCI
C19711	Professional_...	Professional ...	ORG Outside Bodies Organization Professional	A formal group of people that constitute an identifiable social unit with a collection of responsibilities defined by one or more shared goals, who coordinate their activities to achieve these goals.	NCI

Figure 3.22 Search Thesaurus dialog box with automatic search results

- If the Concept Name field was blank when you clicked **Search EVS** (for example, if you are adding a new concept), the Search Thesaurus dialog box appears but is blank.



Search: | Search By: Synonyms | Include Retired? ☐ | Results Per Page: 10

Code	Concept Name	Preferred N...	Synonyms	Definition	Source

Figure 3.23 Blank Search Thesaurus dialog box - enter criteria for search

The Search Thesaurus dialog box contains the following items:

— **Search text box:** Use this field to enter your search criteria. You may use an asterisk (*) as a wildcard. The search is not case-specific.

— **Search By drop-down list:** Use this option to select whether you want to search on the Synonym field or Concept Code field in EVS.

— **Include Retired checkbox**: Use this to select whether to include Retired concepts in the search results. Retired concepts are excluded by default, as active concepts are preferred for mapping.

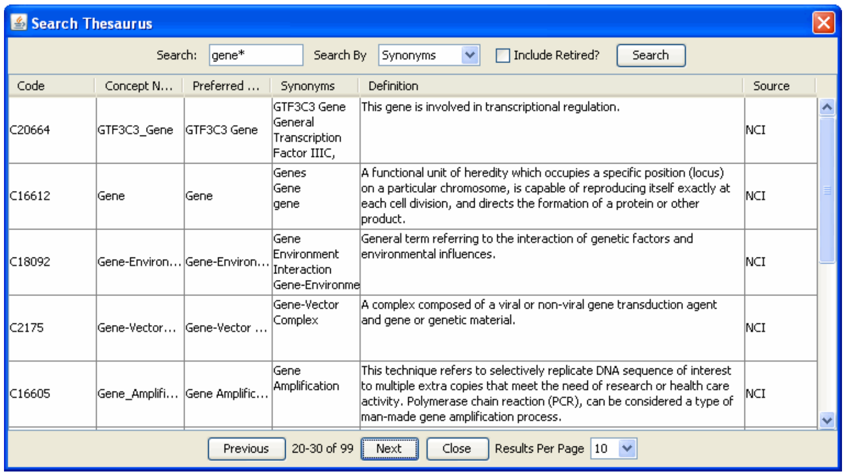
— **Search button**: Click to execute the search based on the criteria you entered.

- — **Close button**: Click to close the search dialog box without mapping any concepts.
- — **Previous and Next buttons**: These buttons only become active if your search results cover more than one page. Use these to page through the result set. Between these buttons is an indicator that lets you know which results are currently displayed along with the total number of items returned.
- — **Results Per Page drop-down list**: Use this option to change how many results appear on each page.

2. If an automatic search was run and the concept you want to map appears in the results, **double-click** the item. The information for the new concept replaces the information that had been in the concept information fields in the SIW. Otherwise, continue with the steps below to search for concepts.
3. Enter your search criteria into the **Search** text box. You have the following options:
   - º Type in the term you are looking for;
   - º Type in part of the term you are looking for along with an asterisk (*) as a wildcard character. This method will return more results but also typically take longer because the result set is more inclusive;
   - º Type in the concept code you want to find. You may NOT use a wildcard to search for concept code. Concept code searches return exact matches only.
4. In the **Search By** drop-down list, select the NCI Thesaurus field to search in: **Synonym** or **Concept Code**.

   Each concept in the NCI Thesaurus has at least one synonym associated with it, and many concepts may share a synonym. For example, a search for "ID" will return multiple results because there are several concepts that have ID listed as a synonym (e.g., Identifier, Idaho, Indonesia). Conversely a search for "Genes" will return "Gene" as its only result because "Genes" only exists as a synonym for "Gene" in the NCI Thesaurus.
5. Check the **Include Retired?** check box if you want the results to include Retired items. Retired concepts are excluded by default.
6. Click **Search**. The results appear below the search options, listed in alphabetical order by concept name. If there are multiple pages of results, the

Previous and **Next** buttons are enabled, allowing you to scroll through the pages.



Code	Concept N...	Preferred ...	Synonyms	Definition	Source
C20664	GTF3C3_Gene	GTF3C3 Gene	GTF3C3 Gene General Transcription Factor IIIC,	This gene is involved in transcriptional regulation.	NCI
C16612	Gene	Gene	Genes Gene gene	A functional unit of heredity which occupies a specific position (locus) on a particular chromosome, is capable of reproducing itself exactly at each cell division, and directs the formation of a protein or other product.	NCI
C18092	Gene-Environ...	Gene-Environ...	Gene Environment Interaction Gene-Environme	General term referring to the interaction of genetic factors and environmental influences.	NCI
C2175	Gene-Vector...	Gene-Vector ...	Gene-Vector Complex	A complex composed of a viral or non-viral gene transduction agent and gene or genetic material.	NCI
C16605	Gene_Amplifi...	Gene Amplific...	Gene Amplification	This technique refers to selectively replicate DNA sequence of interest to multiple extra copies that meet the need of research or health care activity. Polymerase chain reaction (PCR), can be considered a type of man-made gene amplification process.	NCI

Figure 3.24 Search Thesaurus (EVS) dialog box with search results

If needed, you can resize the Search Thesaurus dialog box as well as resizing the columns within the dialog box, to better view the result set details. If you cannot sufficiently resize the box to see the information you need, hold your mouse cursor over each field to display the full text of the field in the mouse-over tool-tip that appears.

7. If the concept you want to map appears in the result set, **double-click** anywhere in the row for that concept. This closes the Search dialog box and enters the information for the selected concept into the concept information fields in the SIW viewer.

8. If the search returns no results (the Search dialog box remains empty) or the concept you want to map does not appear in the result set, alter your search criteria to be more inclusive by using the wildcard or by removing terms from the Search text box if multiple terms appear.

After making changes to the concept information, you can click **Apply** to apply those changes to that element, or select **Edit > Apply to All** to apply your changes to every instance of that element in the model. For more information, see *Applying Changes to All Similar Nodes* on page 70.

Note: The search result list is limited to 100 terms. If your search returns 100 terms, there may be more concepts available that match your criteria. Alter your search results to be more restrictive or specific to the concept information you are looking for.

Editing Concept Information in Curate Mode

Besides adding concepts to elements, EVS curators can also edit the existing concept information for an element. Editing concept information consists of replacing mapped concepts with different concepts, changing the semantic order of mapped concepts, editing the text in the concept information fields, and where necessary, removing concepts from an element's mapping.

Note: If you accidentally make changes to concept information that you did not intend to make or don't want to keep, click **Previous** or **Next** before clicking **Apply**. The SIW prompts you regarding whether or not to save the change. Click **No** and the element's concept information is returned to its original state.

Replacing Mapped Concepts in Curate Mode

Replacing mapped concepts uses the same steps as adding concepts except that the concept information fields are already populated, and if you use the **Search EVS** button to search the NCI Thesaurus, the existing concept name will be used to launch the initial search for concepts.

Some mapped concepts may need to be replaced because the Semantic Connector did not parse the element name properly and performed a search on a term that is not applicable to the element, or returned a result that simply isn't in line with the intent of the element. In this case, one or more of the existing mapped concepts may need to be replaced with more accurate concepts from EVS.

To replace a mapped concept:

1. Select an element to view, and if necessary scroll through the detail view to find the concept you need to replace.
2. Click **Search EVS** located to the right of the Concept Code field. The Search Thesaurus dialog box appears and should contain search results based on the concept name of the concept you are replacing.
3. Since the automated search is probably what mapped the incorrect concept in the first place, change the text in the Search field to return results that better reflect the element concept you are trying to map.
4. Click **Search**.
5. Review the search results. If the concept you want to replace the existing concept with appears in the list, **double-click** anywhere in that row. The new concept information replaces the old concept information.

If the search results do not return any concepts you want to use, revise your search criteria and search again, or consider creating a new concept for the model. For information on creating a new concept for the model, see *Editing Concept Information While Changing the Concept Code* on page 69.

After making changes to the concept information, you can click **Apply** to apply those changes to that element, or select **Edit > Apply to All** to apply your changes to every instance of that element in the model. For more information, see *Applying Changes to All Similar Nodes* on page 70.

Changing the Order of Mapped Concepts in Curate Mode

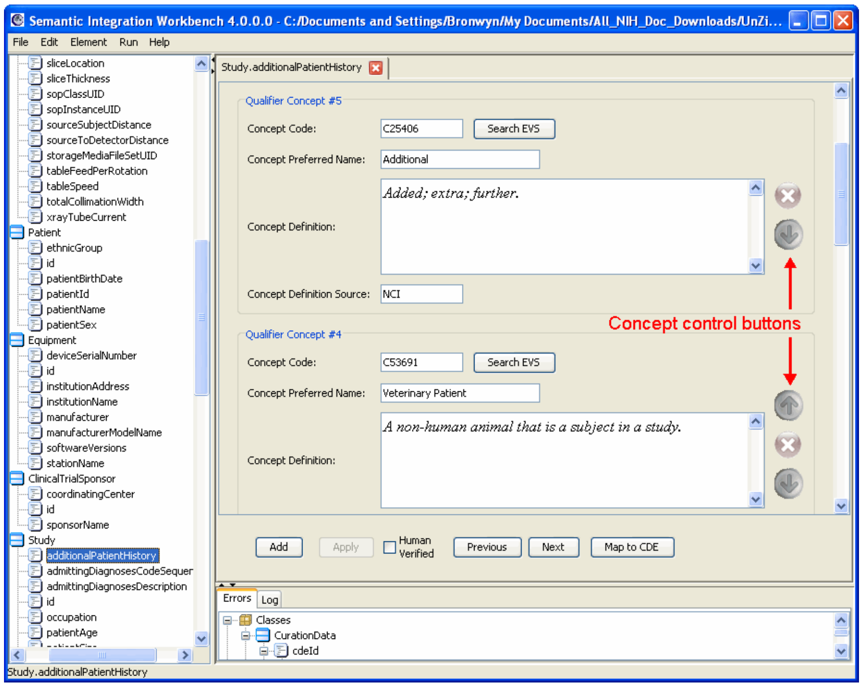
As stated earlier in this section, every element must have a Primary Concept that reflects the basic point or meaning of the item, and then may have Qualifier Concepts that reflect additional details about the element. This is often true of attributes, which can reflect multiple details about the class element they belong to.

The order in which the concepts are mapped is reflected in the order given to those concepts in the SIW. This means that the first concept mapped is always the Primary

Concept, while subsequently mapped concepts become Qualifier concepts with numbers that reflect when they were added (respectively). The order of these Qualifier concepts must reflect the intended semantic meaning of the element to which they are mapped.

The Semantic Connector uses the names of elements to map those elements to EVS concepts, parsing the names into individual parts based on the camel case naming convention. However, while curating the file you may discover that the concepts for an element are not mapped in an order that reflects the purpose for the element. In this case you will want to change the semantic order of the mapped concepts.

Once an element has more than one concept mapped to it, up and/or down arrow buttons appear to the right of the Concept Definition field in the detail view of the SIW viewer.


Semantic Integration Workbench 4.0.0.0 - C:/Documents and Settings/Bronwyn/My Documents/All_NIH_Doc_Downloads/UnZi...
File Edit Element Run Help
Concept control buttons
Add
Apply
Human Verified
Previous
Next
Map to CDE
Search EVS
Errors
Log

sliceLocation
sliceThickness
sopClassUID
sopInstanceUID
sourceSubjectDistance
sourceToDetectorDistance
storageMediaFileSetUID
tableFeedPerRotation
tableSpeed
totalCollimationWidth
xrayTubeCurrent
Patient
ethnicGroup
id
patientBirthDate
patientId
patientName
patientSex
Equipment
deviceSerialNumber
id
institutionAddress
institutionName
manufacturer
manufacturerModelName
softwareVersions
stationName
ClinicalTrialSponsor
coordinatingCenter
id
sponsorName
Study
additionalPatientHistory
admittingDiagnosesCodeSequer
admittingDiagnosesDescription
id
occupation
patientAge

Study.additionalPatientHistory

Qualifier Concept #5

Concept Code: C25406

Concept Preferred Name: Additional

Concept Definition: *Added; extra; further.*

Concept Definition Source: NCI

Qualifier Concept #4

Concept Code: C53691

Concept Preferred Name: Veterinary Patient

Concept Definition: *A non-human animal that is a subject in a study.*

Classes
CurationData
cdeId

Study.additionalPatientHistory

Figure 3.25 An attribute element with multiple concepts and the order control buttons.

The arrow buttons move the concept information either up or down (as indicated by the arrow selected) one position in semantic order for that item. The **X** button removes the concept from the element.

For example, an attribute named “initialDiagnosisDate” has a primary concept of Date, with two qualifying concepts: “Initial” and “Diagnosis”. But for whatever reason, the Qualifier concept #2 is “Diagnosis” and the Qualifier concept #1 is “Initial.” You believe these need to be switched in order to be mapped in proper semantic order.

In this case, clicking the “down” arrow next to Qualifier Concept #2 (“Diagnosis”) causes this concept information to switch places with Qualifier Concept #1 (“Initial”). This changes the semantics of the concept information, though the element itself is still mapped to those EVS concepts. (Be sure to click **Apply** to apply your change.)

The ability to rearrange the semantic order for concepts is most useful if you have to add or replace mapped concepts for the element. In that case, you can simply use the

Search EVS button to find and add/replace concept information in any order you like, and then use the arrows to put the mapped concepts into the proper order, based on the attribute name and the UML element description provided by the model owner.

To change the semantic position of a concept:

1. Select an element from the navigation tree, and if necessary scroll through the detail view to find the concept you need to move.
2. Click the **Up Arrow** or **Down Arrow** located to the right of the Concept Definition field.

Notice that the concept is not simply moved; the concept switches places with the concept located next to it. This means that you will probably have to repeat these steps multiple times to get the concepts into the proper semantic order, moving concepts up and down in the list as necessary.

After making changes to the concept information, you can click **Apply** to apply those changes to that element, or select **Edit > Apply to All** to apply your changes to every instance of that element in the model. For more information, see *Applying Changes to All Similar Nodes* on page 70.

Deleting Mapped Concepts in Curate Mode

The Semantic Connector will typically map multiple concepts to an element when it finds multiple matching names for an element. In this case you will need to remove the mapped concepts that do not apply to that element.

For example, if the model contains a class or attribute with a name “id” it is quite likely that the Semantic Connector will map several different concepts to that element, including “identifier,” “Idaho” and “Indonesia” because the Semantic Connector cannot know what the intent of the element is. Therefore it selects and maps all matches it finds.

To remove a concept from the element:

- Click the **X** located next to the Concept Description field of the concept you want to remove. The concept is deleted.

After making changes to the concept information, you can click **Apply** to apply those changes to that element, or select **Edit > Apply to All** to apply your changes to every instance of that element in the model. For more information, see *Applying Changes to All Similar Nodes* on page 70.

Note: If you accidentally delete a concept that you did not intend to remove, click **Previous** or **Next** before clicking **Apply**. The SIW prompts you regarding whether or not to save the change. Click **No** and the concept information is returned to its original state. Please note that this “undo” action does NOT apply if you select **Apply to All**.

Editing Concept Information Without Changing the Concept Code

Within a model, two concepts with the same concept code must have the same concept name, definition and definition source. This means that if you change any of the properties of a concept without also changing the concept code, the SIW applies the property changes you made to *all* other elements within the model that use the same concept code.

For example, two classes “Person” and “Animal” have the “Name” attribute in common, because both elements have names. The “Name” attribute is annotated with the following concept information:

- `conceptCode: C25192`
- `conceptPreferredName: Name`
- `conceptDefinition: A word or group of words indicating the identity of a person usually consisting of ...`
- `conceptDefinitionSource: NCI Thesaurus`

You change the concept definition to state, “A term consisting of a word or group of words that indicates the specific identity of the item to which the term is applied.” However, you do not change the concept code. When you click **Apply**, the change you made is also made to *every instance of that concept in the model*.

The SIW does this for two reasons: first, concept code is a unique identifier, and everywhere a term in the model is identified with that code, it must reflect the same information. Second, if some aspect of a concept (like definition) within EVS changes for a particular term, that change needs to be applied across all systems (which is the point of semantic integration). Automating this type of change makes curation of XMI files a much more efficient process.

Editing Concept Information While Changing the Concept Code

Since the concept code is a unique identifier, if you change some aspect of the concept information for an element, but you don’t want that change to cascade to all instances of that concept, you must also change the concept code. Changing concept information while also changing the concept code essentially creates a new EVS concept. This is useful if the model contains an element that while is similarly named to other elements, does not map neatly to any existing EVS concepts and therefore requires its own description and definition.

If you make changes to the concept information and change the concept code to a new one, only the element you are editing is modified (as long as you are using a code that is *not* used elsewhere in the model). Those changes do not cascade to other elements.

For example, the model has a class named “Animal” that contains an attribute called “Breed.” The Semantic Connector mapped this to an EVS concept whose definition reflects the noun form of the term “breed” rather than the verb form. This model intends for the attribute to reflect the act of breeding instead of a type of animal.

During curation, you (the curator) can tell the intent of the attribute because of the UML definition provided by the model owner (see why these are important?). So you change the definition for the concept and also change the concept code so that your changes are only applied to this element. Once entered into EVS, there will be two EVS concepts called “Breed” with two different concept codes and two different concept definitions.

Caution: If you do change the concept code for a mapped concept, be sure you are using a new code and *not* an existing one. If you change the code to one that is in use in the model, your changes will cascade to all elements mapped to the concept with that code.

You may find that you need to apply your concept changes to all similar elements in the model. For information on this feature, see *Applying Changes to All Similar Nodes* below.

Applying Changes to All Similar Nodes

In some cases, an attribute is reused several times across the model. A common example is the attribute “id.” Typically, this attribute represents the same type of information everywhere it is used across the model. As noted in an earlier example, the term “id” may apply to a list of different types of information, and therefore the Semantic Connector may have mapped several different concepts to this element. These extra concept mappings will need to be removed from every instance of the “id” attribute.

The **Apply to All** command, located in the Edit menu, allows you to apply the changes made to an element to all similar items in the model.

Apply to All is used in place of **Apply**. If you click **Apply** first, the **Apply to All** option is grayed out and no longer available.

Using the example of the “id” attribute, in the model, “id” occurs 27 times, and 25 of those times, refers to “Identifier.” All 27 instances of “id” have been mapped to the concept “Identifier” but they have also been mapped to “Idaho” and “Indonesia” by the Semantic Connector. You can remove these unnecessary concepts from one of the “id” attributes in the model, then select **Edit > Apply to All**. This removes those concepts from all instances of the attribute “id.”

You can then go back and manually update or replace the concept information for the two instances of “id” that refer to something other than “Identifier.”

The **Apply to All** feature is also useful if you are adding concepts to an element or reordering the concepts in an element and need to apply those same changes to all or most of the other instances of the element in the model. For information on adding concepts, see *Adding EVS Concepts to an Element in Curate Mode* on page 60. For information on editing concept information, see *Editing Concept Information in Curate Mode* on page 65.

Caution: If you use **Apply to All** you will no longer have the ability to “undo” your changes by selecting a different element in the model and then selecting not to apply the changes. Once **Apply to All** is used, all similar elements are changed to reflect the changes made in the currently active element. If you need to undo those changes, you can *exit the SIW without saving the changes to the file*, and then reopen the file. You will, however, lose all of your changes since you last saved the file.

Remember that applying changes is not the same as saving the file. You must still use the **File > Save** or **File > Save As** commands to save your changes to the file. For more information, see *Saving Changes to a File* on page 39.

Validating XMI File Concept Information Against EVS

The **Validate Concepts** option, located under the Run menu, performs a comparison of the Concept Code, Concept Name and Concept Definition fields for each mapped element in the currently open XMI file to the “official” information located in EVS. This allows you to easily see where there are differences or where there is concept information in the model that is not resident in EVS.

To Validate the XMI concepts against EVS concepts:

1. With the annotated XMI file open in Review Annotated XMI File mode, select **Run > Validate Concepts**.
2. A confirmation dialog box appears, informing you that the validation process may take some time to complete, and asking if you want to continue. Click **Yes**.
3. An empty Validate Concepts window appears, with a progress bar at the bottom informing you of the status of the validation process.

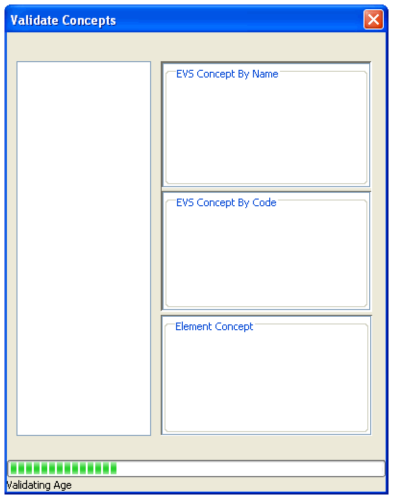


Figure 3.26 Validate Concepts window showing progress of the validation

   When the validation process completes, the progress bar spans the bottom of the window, and a status of **Done** appears below the progress bar. In addition, the left side of the window will list all elements in the XMI file that contain concept information that differs from what appears in EVS.

   If the left side of the window is empty, there are no disparities between the XMI file and EVS.

4. Select one of the elements from the list on the left side of the Validate Concepts window. This populates the other areas of the window with concept name, concept code and concept definition information as follows:
   - **EVS Concept by Name** - The area located on the top-right of the Validate Concepts window shows the entry in EVS for the concept name that matches the concept name used for the selected element in the XMI file.

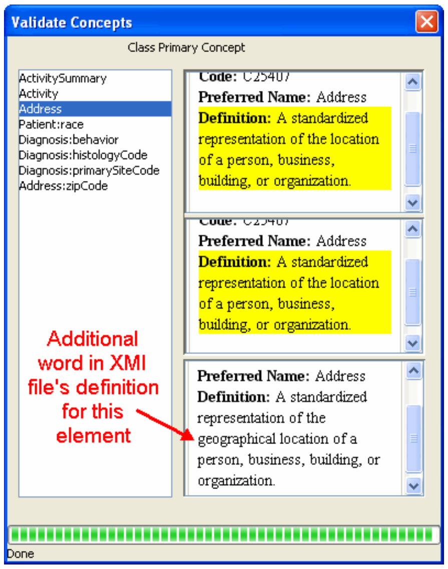
     If this area is empty, it means that no match was found in EVS for the concept name listed for the selected element.
   - **EVS Concept by Code** - The area located in the middle-right of the Validate Concepts window shows the entry in EVS for the concept code that matches the concept code used for the selected element in the XMI file.

     If this area is empty, it means that no match was found in EVS for the concept code listed for the selected element.

- **Element Concept** - The area located on the bottom-right of the Validate Concepts window shows the concept information resident in the XMI file that corresponds with the EVS concept(s) appearing in the upper two boxes. This allows you to directly compare what is in the XMI file with the corresponding information in EVS.

In addition to these informational areas, the top of the Validate Concepts window shows the type of element and type of disparate concept currently selected (e.g., Class Primary Concept or Attribute Qualifier Concept #2).

The Validate Concepts window highlights the fields where there are differences between the information in EVS and in the XMI file.



Validate Concepts

Class Primary Concept

ActivitySummary
Activity
Address
Patient:race
Diagnosis:behavior
Diagnosis:histologyCode
Diagnosis:primarySiteCode
Address:zipCode

Code: C25407
Preferred Name: Address
Definition: A standardized representation of the location of a person, business, building, or organization.

Code: C25407
Preferred Name: Address
Definition: A standardized representation of the location of a person, business, building, or organization.

Preferred Name: Address
Definition: A standardized representation of the geographical location of a person, business, building, or organization.

Done

Figure 3.27 Validate Concepts window showing a difference in concept definition

You can make changes to the information in the Element Concept section of the Validate Concepts window, which makes those changes to the element in the XMI file.

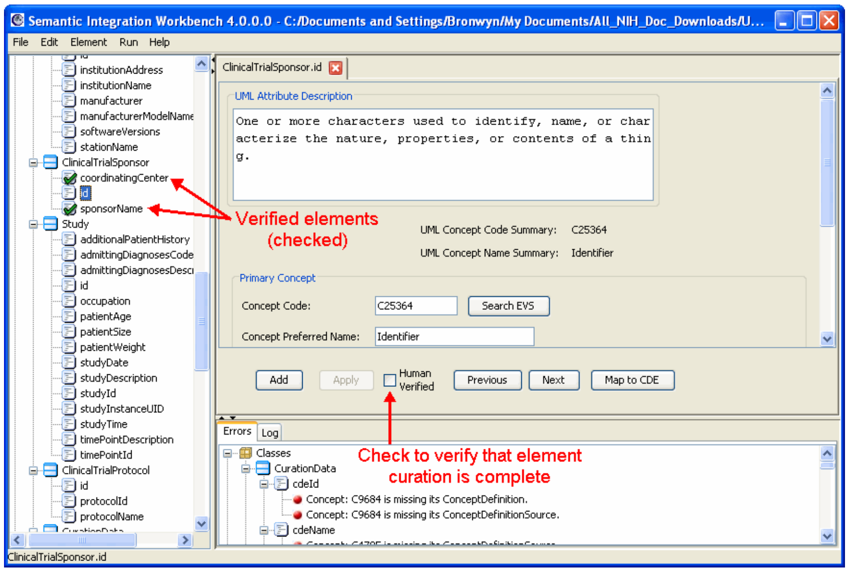
In addition, when you select an element in the Validate Concepts window, the same item is also selected in the navigation tree of the main SIW viewer window. This allows you to review all of the other concept information for that element if needed.

Note: Typically the EVS information in the top two boxes of the Validate Concepts window is the same, however it is possible that the element in the XMI file has a EVS concept code that differs from the EVS concept name associated with that code. In this case, the search performed by the validation would return different results for these areas of the window. The Validate Concepts window will highlight these differences.

Verifying the Curated XMI File

Once you have reviewed and updated the concept information in the file to your satisfaction, you can use the Human Verified checkbox to mark that curation for this element is complete. As is logical, the Human Verified checkbox is not available for any element that does not contain concept information or which is not already mapped to a CDE or OC.

Once an element has been checked as Human Verified, a checkmark also appears next to the element in the navigation tree. This makes it very easy for you to identify which elements still need to be reviewed and verified.



One or more characters used to identify, name, or characterize the nature, properties, or contents of a thing.

UML Concept Code Summary: C25364
UML Concept Name Summary: Identifier
Concept Code: C25364
Concept Preferred Name: Identifier

Concept: C9684 is missing its ConceptDefinition.
Concept: C9684 is missing its ConceptDefinitionSource.

Figure 3.28 SIW Viewer in Curate XMI File mode, with human verified elements

After the file is fully curated, you must mark *all* elements as **Human Verified**, to mark the elements in the navigation tree are marked with a checkmark. This includes the elements that you either mapped or verified the mapping of concept information for, and the elements that are already mapped to CDEs or OCs.

To verify an element:

1. Select an element from the navigation tree.
2. Review the information in the detail view to verify that all of the necessary information is mapped to the element.
3. Click the **Human Verified** checkbox to enable it. One of the following occurs, depending on the element you are verifying:
    - If you have selected a class element and there are un-verified attributes associated with that class, the next item in the tree is automatically selected for you to verify.
    - If you have selected an attribute, a checkmark appears on the attribute in the navigation tree and the next item in the tree is automatically selected.
    - If you have selected the last attribute in the class to be verified (all others have already been verified), and the class has already been verified, a checkmark appears on the attribute you are verifying and on the class element, and the next item in the tree is automatically selected.

The automatic movement of the SIW to the next element during element verification is simply a navigational aid, so that you do not have to continually click **Next** to move to the next item. Once an element is verified, the SIW assumes that you are finished with that item.

Once you have reviewed all items in the file, and cleared all of the errors listed in the Errors tab (which may require multiple launches of the SIW and the XMI file to update the Errors tab), you must save the file using the **File > Save As** command. Curated and verified files should be saved with the term “annotated” appended to the front of the filename. For example, if the original file name was `FirstPass_myModel.xmi` then you would save the file as `Annotated_FirstPass_myModel.xmi`. If the original file had a .uml extension (exported from ArgoUML) then the curated file must also be saved with a .uml extension.

For further instructions on saving a file, see *Saving Changes to a File* on page 39.

The curated file should be ready for Review. The curated XMI file is used as input to the Reviewing an Annotated XMI File step.

Reviewing an Annotated XMI File

The **Review Annotated XMI File** step is performed by the model owner after the EVS curation team finishes annotating the file and returns the annotated file. During this review step in the semantic integration process, the model owner can make corrections and/or simply verify the mappings contained in the annotated XMI file. Users can search EVS for concepts to change the concept mapping for a class or attribute, or users may choose to map UML attributes to existing caDSR value domains or data elements where valid, applicable caDSR items already exist.

The main difference between working with a file in the Review Annotated XMI File mode as opposed to the Review Unannotated XMI File or Curate XMI File mode is that the Review mode shows *all* of the items from the UML model that will be uploaded into caDSR. The Review mode also shows concept errors and other problems with the file that were not relevant to an unannotated file.

When opening a file in the Review Annotated XMI File step, the SIW performs a number of validation checks to ensure that the XMI file can be correctly transformed into caDSR metadata. For each UML class and attribute in the file, the SIW checks for the presence of at least one of each of the following:

- Concept code
- Concept name
- Concept definition
- Concept definition source
- Valid datatype

Furthermore, when the file passed the Curate XMI File step, each element was checked as Human Verified. When the annotated file is opened in Review Annotated XMI File mode, those checkmarks are reset so that the model owner can review each element in the file and when satisfied that the element is complete, check the element as Model Owner Verified.

If any of the elements in the file are missing any of the required mapping items, the SIW lists an error in the Errors tab, allowing the model owner to review the error and make changes to the element mapping as appropriate.

Note: All errors must be corrected before a model can be loaded to the caDSR.

***Reminder about the Errors tab*—**The SIW generates the Errors tab information when the file is opened; it cannot update the error listing dynamically. This means that as you make changes, you will need to save the file, exit the SIW, then reopen the saved file in the Review Annotated XMI File mode to see an updated error listing.

Input: The Annotated XMI file as curated by the EVS concept curation team. The file is returned to the model owner with the term "Annotated" added to the beginning of the file name.

Example: `Annotated_FirstPass_myModel.xmi` or `Annotated_FirstPass_Roundtrip_myModel.xmi` for EA generated models. For ArgoUML-generated models, the file names would end with a `.uml` extension.

Output: An Approved Annotated XMI file. When you save the file, you should append it with the term "approved" at the front of the filename. For example: `Approved_Annotated_FirstPass_myModel.xmi` or `Approved_annotated_FirstPass_Roundtrip_myModel.xmi`. For ArgoUML generated models, the file names would end with a `.uml` extension.

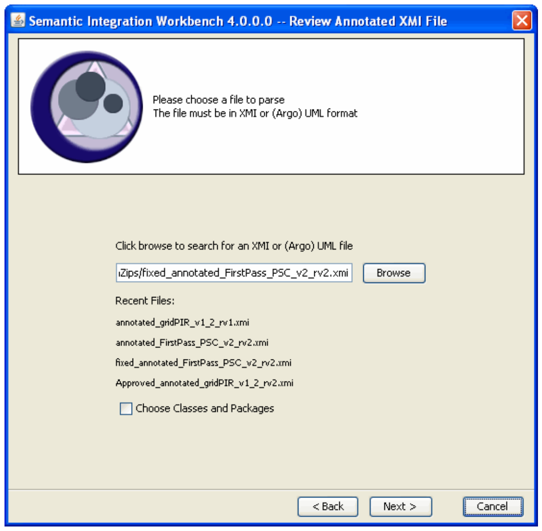
Note: If you need to return the file to the EVS curation team for an additional curation pass at the file after review, you may choose to save the file with the term "fixed" appended to the front, rather than "approved." In addition, you may also want to employ the use of revision numbering in the file name, so that you can delineate between the different versions of processed files (e.g., fixed_r2_annotated_FirstPass_myModel.xmi).

The end result of the Review XMI file step is a file that contains no errors, and where all of the elements in the file have been checked as "Model Owner Verified." Once you have reviewed the concept mappings for each element in the file and mapped elements to caDSR CDEs where appropriate, you can then finally verify all of the elements in the file and save the file as an "Approved Annotated" version of the XMI file. The approved annotated XMI file is the file that will be passed on to NCICB for logging into CVS and loading to caDSR.

To Review an Annotated XMI file:

1. Be sure you have saved the annotated (curated) XMI file you received from the EVS curators.
2. Open the SIW URL: http:\\cadsrsiw.nci.nih.gov.

3. From the SIW Welcome screen, select option **5. Review Annotated XMI File (Model Owner)** and click **Next**. The file selection screen appears.


Semantic Integration Workbench 4.0.0.0 -- Review Annotated XMI File
Browse
< Back
Next >
Cancel

Please choose a file to parse
The file must be in XMI or (Argo) UML format

Click browse to search for an XMI or (Argo) UML file

iZips/fixed_annotated_FirstPass_PSC_v2_rv2.xmi

Recent Files:

annotated_gridPIR_v1_2_rv1.xmi

annotated_FirstPass_PSC_v2_rv2.xmi

fixed_annotated_FirstPass_PSC_v2_rv2.xmi

Approved_annotated_gridPIR_v1_2_rv2.xmi

☐ Choose Classes and Packages

Figure 3.29 Select a file for SIW to parse and display for review

If you have used the SIW before, the last five files used for this option appear in a Recent Files list located below the filename text box.

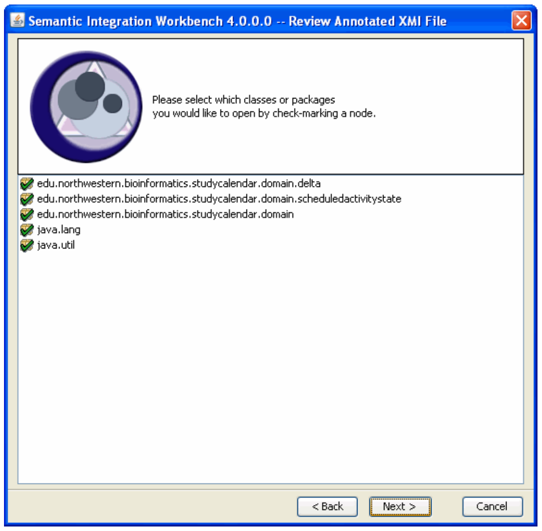
4. Identify the file you want to review. You have the following options:

   - If the file you want to review appears in the **Recent Files** list, click on the file name to populate the filename field with the path/filename.
   - **Type** the full path and filename into the text box.
   - Click **Browse** to browse for and select the file exported from EA or ArgoUML.

Note: The Browse function defaults to selecting (showing) only files with an .xmi file extension. If you are using a .uml file, select **UML Files** from the **Files of Type** drop-down list at the bottom of the Open dialog box.

5. If you want to review only portions of the file, enable the **Choose Classes and Packages** checkbox. Otherwise the SIW will parse the entire file for review.

6. Click **Next**. If you enabled the **Choose Classes and Packages** checkbox, the package selection screen appears, listing all of the packages in the file.


Semantic Integration Workbench 4.0.0.0 -- Review Annotated XMI File
< Back
Next >
Cancel

Please select which classes or packages
you would like to open by check-marking a node.

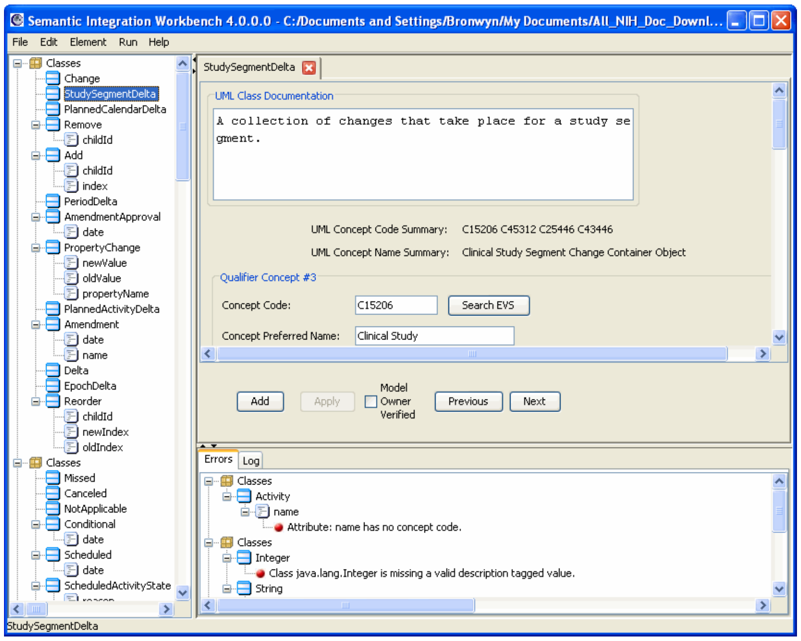
- edu.northwestern.bioinformatics.studycalendar.domain.delta
- edu.northwestern.bioinformatics.studycalendar.domain.scheduledactivitystate
- edu.northwestern.bioinformatics.studycalendar.domain
- java.lang
- java.util

Figure 3.30 Package selection screen - lists the packages resident in the file

7. All packages listed are checked by default. Click a **checkmark** to open a list of all items contained within the package, and then click the checkmarks next to any items you do not want the SIW to parse for review. Then click **Next**.

The SIW displays the file selection screen with a progress bar at the bottom showing the current action and overall progress. How long this process will take is determined by the size of the XMI file or packages you are using as well as the number of errors or warnings the SIW must generate.

When the SIW is finished parsing the annotated file, the SIW viewer appears showing the selected file (or selected packages/classes).


Semantic Integration Workbench 4.0.0.0 - C:/Documents and Settings/Bronwyn/My Documents/All_NIH_Doc_Downl...
File
Edit
Element
Run
Help
StudySegmentDelta
Add
Apply
Previous
Next
Search EVS
Errors
Log
StudySegmentDelta

```
Classes
  Change
  StudySegmentDelta
  PlannedCalendarDelta
  Remove
    childId
  Add
    childId
    index
  PeriodDelta
  AmendmentApproval
    date
  PropertyChange
    newValue
    oldValue
    propertyName
  PlannedActivityDelta
  Amendment
    date
    name
  Delta
  EpochDelta
  Reorder
    childId
    newIndex
    oldIndex
Classes
  Missed
  Canceled
  NotApplicable
  Conditional
    date
  Scheduled
    date
  ScheduledActivityState
```

UML Class Documentation

A collection of changes that take place for a study se
gment.

UML Concept Code Summary: C15206 C45312 C25446 C43446

UML Concept Name Summary: Clinical Study Segment Change Container Object

Qualifier Concept #3

Concept Code: C15206

Concept Preferred Name: Clinical Study

☐ Model Owner Verified

```
Classes
  Activity
    name
      Attribute: name has no concept code.
Classes
  Integer
    Class java.lang.Integer is missing a valid description tagged value.
  String
```

Figure 3.31 SIW Viewer - showing an Annotated XMI file with errors

Use the SIW viewer to review the mappings for each element, and where listed, the errors encountered in the file, to determine what changes need to be made in the file before it can be considered "approved" and then uploaded to caDSR.

If necessary, see *Using the SIW Viewer* on page 27 for a basic overview on navigating the SIW viewer and on using the **Previous**, **Next**, **Add**, **Apply** and **Search EVS** buttons.

You are also *strongly* encouraged to review the *Editing Concept Information in Curate Mode* section beginning on page 65. This section provides detailed information on making changes to the concept information of your file, including searching EVS, replacing mapped concepts, reordering concepts, and removing concepts for an element.

Changing Concept Information in Review Mode

While reviewing the annotated file you received from the EVS curation team, you may find that you need to make changes to the concept information mapped to some of the elements. The information in this section provides basic procedures for how to add, replace, reorder and remove concept information for an element. For more detailed information on these features, see *Adding EVS Concepts to an Element in Curate Mode* on page 60 and *Editing Concept Information in Curate Mode* on page 65.

If you do make changes to the concept information in your model, you may want to validate the concepts in your XMI file against those in EVS. The **Validate Concepts** option, located under the **Run** menu, performs a comparison of the concept information in your XMI file against the concept information in EVS. For more information on using this feature, see *Validating XMI File Concept Information Against EVS* on page 70.

Note: If you accidentally make changes to concept information that you did not intend to make or don't want to keep, click **Previous** or **Next** *before* clicking **Apply**. The SIW prompts you regarding whether or not to save the change. Click **No** and the element's concept information is returned to its original state.

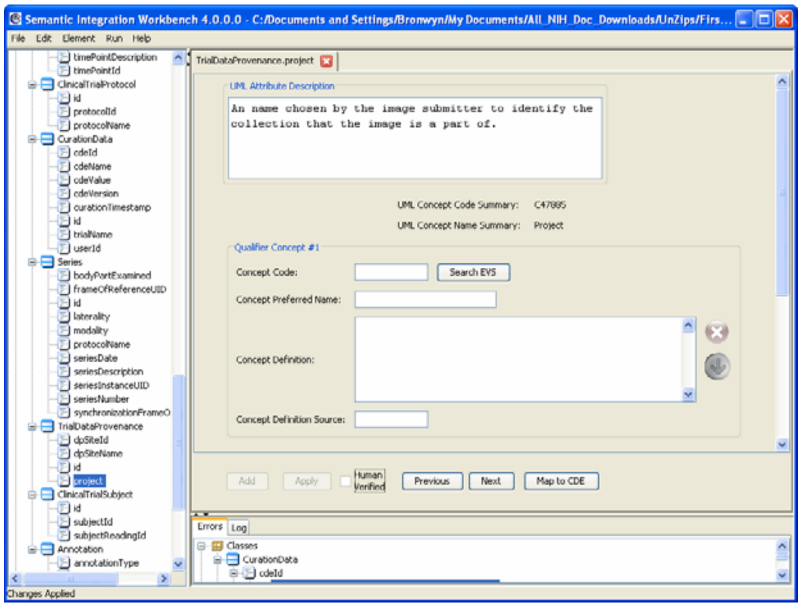
Adding EVS Concepts to an Element in Review Mode

Even though your file has been curated, you may find that it requires further addition of Qualifier concepts.

To add a concept:

1. In the SIW viewer, select an element from the navigation tree.
2. In the detail view, click **Add**. A set of empty concept fields appears, along with a **Search EVS** button.

   Note: If you have changed your viewer Preferences to Display Primary Concept First, the new concept fields will appear below the existing mapped concepts, meaning you may need to scroll down in the detail view to see the new fields. See *Setting Viewer Preferences* on page 36 for more information.



An name chosen by the image submitter to identify the collection that the image is a part of.

UML Concept Code Summary: C47885

UML Concept Name Summary: Project

Figure 3.32 SIW Viewer in Curate XMI mode, showing new blank concept fields.

3. Click **Search EVS** and perform a search for the appropriate concept to map.
4. Enter the item you want to search for into the Search text box, select whether you are searching for a **Synonym** or **Concept Code**, then click **Search**.
5. When the concept you want to add appears in the search result set, **double-click on the item** to populate the concept fields with the information. For more details on using the EVS Search to populate the concept fields, see *Searching EVS for Concept Information* on page 62.
6. When the concept fields are complete, click **Apply**. This applies the new concept to the element. (You will still have to save the file after all of your changes are made. See *Saving Changes to a File* on page 39.
7. If necessary, after applying your changes, you can click **Add** again and complete the above steps to add another concept to this element.

Replacing Mapped Concepts in Review Mode

Some mapped concepts may need to be replaced because the concept information mapped to them is not exactly what you intended for the item. In this case, you can search EVS for a better match and replace the existing concept.

Replacing mapped concepts is the same as adding concepts to an element, except that the concept information fields are already populated. Unless you have changed your default viewer Preferences, clicking **Search EVS** automatically launches a search of the NCI Thesaurus using the existing concept name of the item. See *Setting Viewer Preferences* on page 36 for more information.

Note: If you cannot find a matching concept in EVS, contact the EVS curation team to work with them on finding an existing concept or creating a new one.

To replace a mapped concept:

1. Select an element to view, and if necessary scroll through the detail view to find the concept you want to replace.
2. Click **Search EVS**, located to the right of the Concept Code field of the concept you want to replace. The Search Thesaurus dialog box appears and should contain search results based on the text in the Concept Name field of the concept you are replacing.
3. Since the automated search may be how the incorrect concept was mapped in the first place, you will probably want to change the text in the Search field to return results that better reflect the element concept you are trying to map.
4. Click **Search**.
5. Review the search results. If the concept you want to use appears in the list, **double-click** anywhere in that row. The new concept information replaces the old concept information.

If the search results do not return any concepts you want to use, revise your search criteria (using a wildcard [*] if necessary) and search again. If you cannot find a matching concept in EVS, contact the EVS curation team to work with them on finding an existing concept or creating a new one.

After making changes to the concept information, click **Apply** to apply the changes to that element. If you attempt to navigate away from the element without applying your changes, the SIW prompts you to apply or discard your changes.

Changing the Order of Mapped Concepts in Review Mode

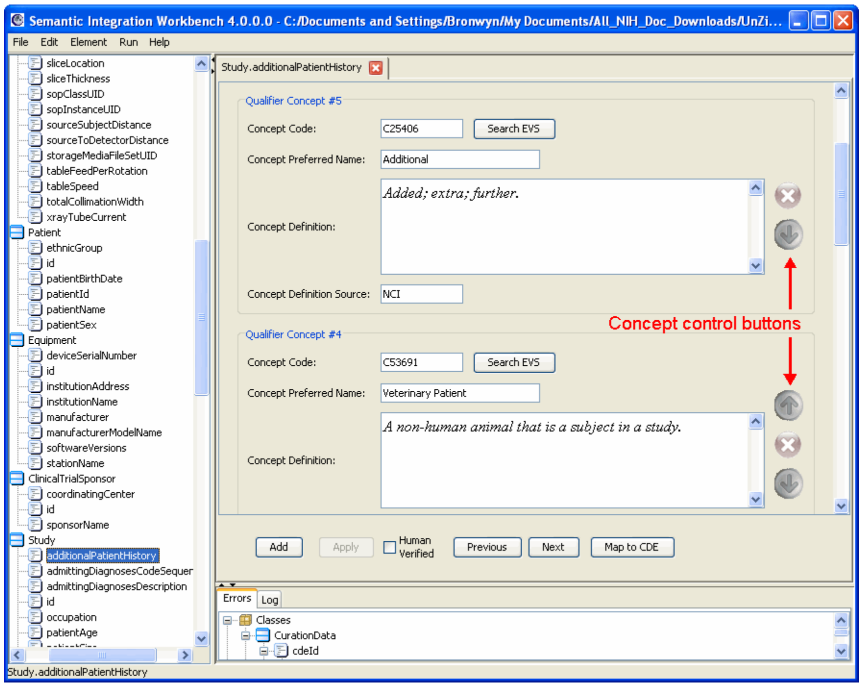
Every element must have a Primary Concept that reflects the basic point or meaning of the item, and then may have Qualifier Concepts that reflect additional details about the element. This is often true of attributes, which can reflect multiple details about the class element to which they belong.

The order in which the concepts are mapped reflects the order given to those concepts in the SIW. This means that the first concept mapped is always the Primary Concept, while subsequently mapped concepts become Qualifier concepts with numbers that reflect when they were added (respectively). The order of these Qualifier concepts must reflect the intentional semantic meaning of the element to which they are mapped.

However, you may discover that the concepts for an element are not mapped in an order that reflects your purpose for the element. In this case you will want to change the semantic order of the mapped concepts.

For more information on the semantic ordering of mapped concepts, see *Changing the Order of Mapped Concepts in Curate Mode* on page 66.

Once an element has more than one concept mapped to it, up and/or down arrow buttons appear to the right of the Concept Definition field in the detail view of the SIW viewer.



Qualifier Concept #5
Concept Code: C25406
Concept Preferred Name: Additional
Concept Definition: *Added; extra; further.*
Concept Definition Source: NCI

Qualifier Concept #4
Concept Code: C53691
Concept Preferred Name: Veterinary Patient
Concept Definition: *A non-human animal that is a subject in a study.*

Figure 3.33 An attribute element with multiple concepts and the order control buttons.

The arrow buttons move the concept information either up or down (as indicated by the arrow selected) one position in semantic order for that item. The **X** button removes the concept from the element.

The ability to rearrange the semantic order for concepts is most useful if you have to add or replace mapped concepts for the element. In that case, you can use the **Search EVS** button to find and add/replace concept information in any order you like, and then use the arrows to put the mapped concepts into the proper order, based on the intended purpose of the element in the model.

To change the semantic position of a concept:

1. Select an element to view, and if necessary scroll through the detail view to find the concept you need to move.
2. Click the **Up Arrow** or **Down Arrow** located to the right of the Concept Definition field.

Notice that the concept is not simply moved; the concept switches places with the concept located next to it. This means that you will probably have to repeat these steps multiple times to get the concepts into the proper semantic order, moving concepts up and down in the list as necessary.

When finished, look at the **Summary Concept Name** listed for the element, and be sure it reflects your intended meaning or use of the element within the model.

After making changes to the concept information, click **Apply** to apply the changes to that element.

Deleting Mapped Concepts in Review Mode

If you have added unnecessary or inaccurate concepts to an element, or find that the curation process left concepts that should not be mapped to an element, you can delete them. However you may only remove concepts if there are multiple concepts mapped to an element (because each element must have a Primary concept). If your element contains only one concept, you must add a new concept before you can delete the existing one. See *Adding EVS Concepts to an Element in Review Mode* on page 78.

To remove a concept from an element:

- Click the **X** located next to the Concept Description field of the concept you want to remove. The concept is deleted.

After making changes to the concept information, click **Apply** to apply those changes to the element.

Note: If you accidentally delete a concept that you did not intend to remove, click **Previous** or **Next** before clicking **Apply**. The SIW prompts you regarding whether or not to save the change. Click **No** and the concept information is returned to its original state. Please note that this "undo" action does NOT apply if you select to **Apply to All**. In that instance you must exit the SIW without saving your changes and re-open the file. You will lose all of your change since the last time you saved the file.

Mapping UML Attributes to caDSR CDEs

Each UML attribute in your model can either be mapped to one or more EVS concepts or mapped to a caDSR CDE. Mapping an attribute to a CDE automatically maps the class element to the caDSR OC of the CDE and the datatype for the attribute to the value domain of the CDE. For more information on the relationship between UML model elements and caDSR CDEs, see *Terms You Should Know* on page 10 and *Using Pre-mapped CDEs (Common Data Elements)* on page 101.

While you are reviewing your annotated file, you may find that you can map some of the attributes in your model to CDEs. This mapping provides direct connectors between the items in your model and caDSR metadata, and improves the semantic integration of your model with other caCORE-compatible systems.

When viewing an attribute in Review Annotated XMI File mode, for any attribute not already mapped to a CDE, the detail view provides a **Map to CDE** button. Clicking **Map to CDE** displays a different detail view including a **Search Data Element** button that allows you to search the caDSR for CDEs to map to the selected attribute.

When you click **Map to CDE**, notice that it turns into a **Map to Concepts** button. This allows you to toggle between the CDE mapping and EVS concept mapping views for the selected attribute. Once you have applied to mapping of the attribute to a CDE however, the concept information no longer appears, in favor of the CDE mapping.

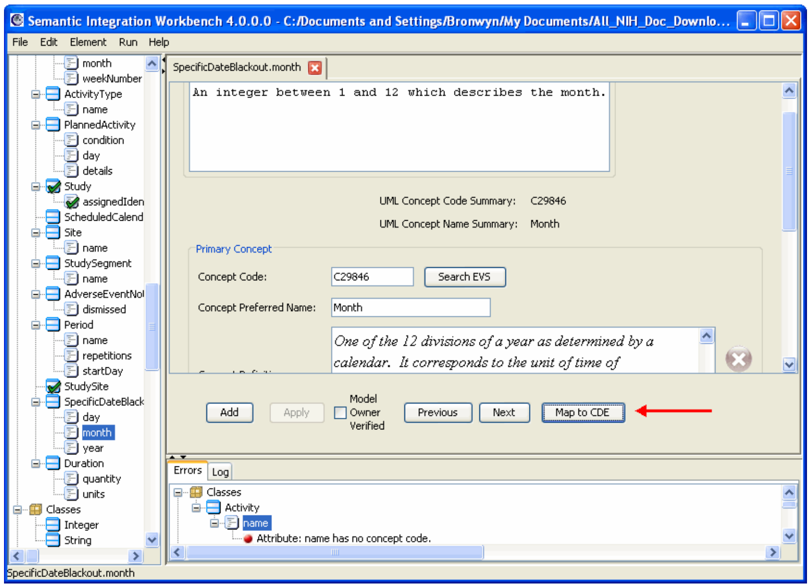
Note: The **Map to CDE** button is not available if the attribute uses a LVD. This is because mapping an attribute to a CDE automatically maps the caDSR value domain as well. For more information on creating and using LVDs for your model instead of caDSR value domains, see *Using Local Value Domains* on page 17.

Mapping a UML Attribute to an Existing Common Data Element

Any attribute in the UML model can be mapped to an existing caDSR CDE. In addition, you can replace a CDE mapping if it is incorrect or needs to be updated to use a newer version of the CDE.

To map an attribute to a CDE:

1. Select an attribute in the navigation tree. If the attribute has not already been mapped to a CDE, concept information appears in the detail view along with a **Map to CDE** button. If the element is already mapped to a CDE, skip to Step 3


Semantic Integration Workbench 4.0.0.0 - C:/Documents and Settings/Bronwyn/My Documents/All_NIH_Doc_Downlo...
File Edit Element Run Help
SpecificDateBlackout.month
Add
Apply
Model Owner Verified
Previous
Next
Map to CDE
Search EVS
Primary Concept
Errors
Log
SpecificDateBlackout.month

An integer between 1 and 12 which describes the month.

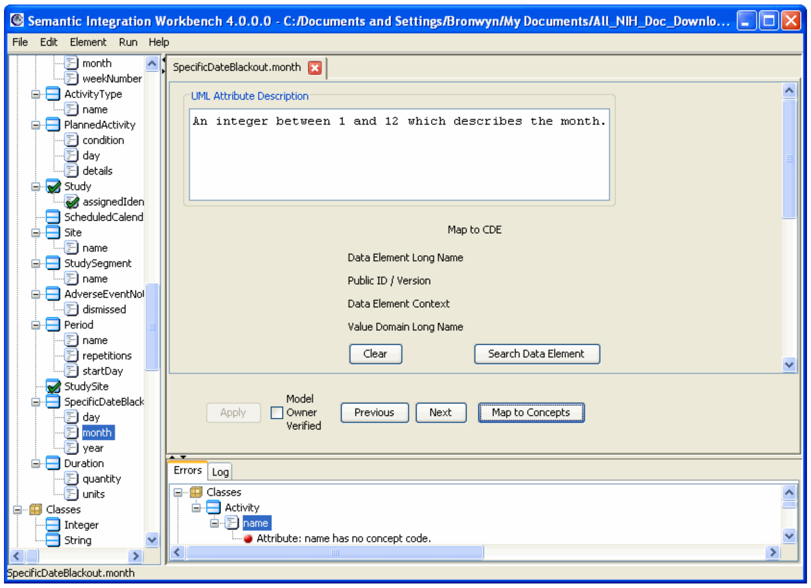
UML Concept Code Summary: C29846
UML Concept Name Summary: Month

Concept Code: C29846
Concept Preferred Name: Month
One of the 12 divisions of a year as determined by a calendar. It corresponds to the unit of time of

Classes / Activity / name
Attribute: name has no concept code.

Figure 3.34 SIW Viewer Detail View showing Map to CDE button

2. Click **Map to CDE**. The detail view changes to show empty data element fields. In addition two new buttons appear: **Search Data Element** and **Clear**. Also notice that the **Map to CDE** button has changed to **Map to Concepts**.


Semantic Integration Workbench 4.0.0.0 - C:/Documents and Settings/Bronwyn/My Documents/All_NIH_Doc_Downlo...
File Edit Element Run Help
SpecificDateBlackout.month
UML Attribute Description
Map to CDE
Data Element Long Name
Public ID / Version
Data Element Context
Value Domain Long Name
Clear
Search Data Element
Apply
Model Owner Verified
Previous
Next
Map to Concepts
Errors
Log
SpecificDateBlackout.month

An integer between 1 and 12 which describes the month.

Classes / Activity / name
Attribute: name has no concept code.

Figure 3.35 Detail View after the Map to CDE button is clicked

3. Click **Search for Data Element**. This opens the Search for Data Element dialog box, allowing you to search the caDSR for CDEs to map your attribute.

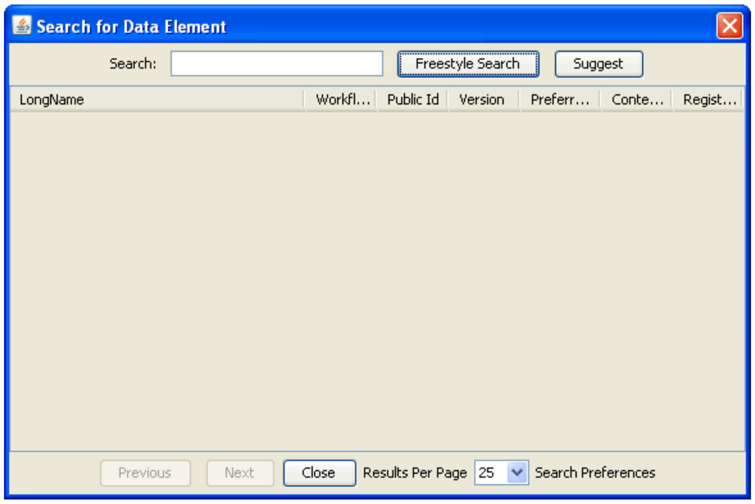


Figure 3.36 Search for Data Element dialog box

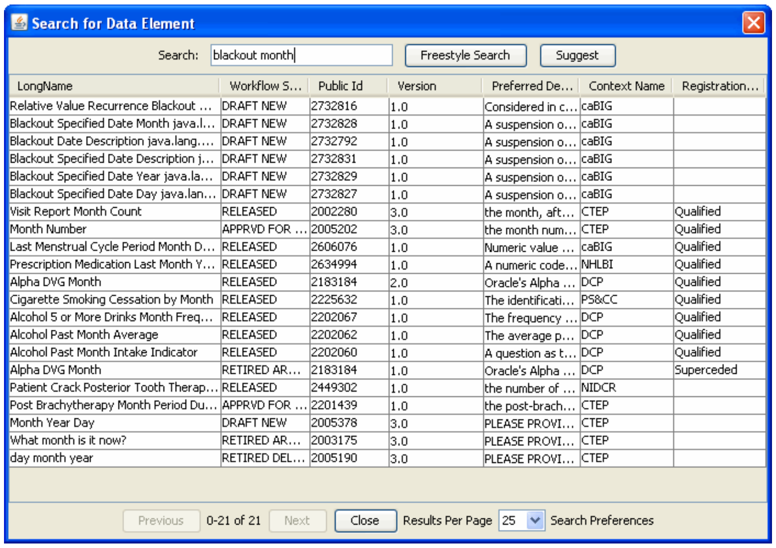
Note: If you have conducted a previous search for a data element during the current session, the Search for Data Element dialog box will be populated with the previous search information/results.

4. In the Search for Data Element dialog box you have two choices:

   - Enter a term in the Search field and click **Freestyle Search**. This search takes advantage of the Freestyle search engine which searches a variety of fields in the caDSR for the term entered (http://freestyle.nci.nih.gov/freestyle/do/search). You may use the asterisk (*) as a wildcard character.

   - Click **Suggest**. This searches the caDSR for any element with an alternate name that matches the attribute name you are attempting to map. **Suggest** uses only the attribute name ignoring any information in the Search text box.

   The results appear below the search options.



Search: blackout month

LongName	Workflow S...	Public Id	Version	Preferred De...	Context Name	Registration...
Relative Value Recurrence Blackout ...	DRAFT NEW	2732816	1.0	Considered in c...	caBIG	
Blackout Specified Date Month java.l...	DRAFT NEW	2732828	1.0	A suspension o...	caBIG	
Blackout Date Description java.lang....	DRAFT NEW	2732792	1.0	A suspension o...	caBIG	
Blackout Specified Date Description j...	DRAFT NEW	2732831	1.0	A suspension o...	caBIG	
Blackout Specified Date Year java.la...	DRAFT NEW	2732829	1.0	A suspension o...	caBIG	
Blackout Specified Date Day java.lan...	DRAFT NEW	2732827	1.0	A suspension o...	caBIG	
Visit Report Month Count	RELEASED	2002280	3.0	the month, aft...	CTEP	Qualified
Month Number	APPRVD FOR ...	2005202	3.0	the month num...	CTEP	Qualified
Last Menstrual Cycle Period Month D...	RELEASED	2606076	1.0	Numeric value ...	caBIG	Qualified
Prescription Medication Last Month Y...	RELEASED	2634994	1.0	A numeric code...	NHLBI	Qualified
Alpha DVG Month	RELEASED	2183184	2.0	Oracle's Alpha ...	DCP	Qualified
Cigarette Smoking Cessation by Month	RELEASED	2225632	1.0	The identificati...	PS&CC	Qualified
Alcohol 5 or More Drinks Month Freq...	RELEASED	2202067	1.0	The frequency ...	DCP	Qualified
Alcohol Past Month Average	RELEASED	2202062	1.0	The average p...	DCP	Qualified
Alcohol Past Month Intake Indicator	RELEASED	2202060	1.0	A question as t...	DCP	Qualified
Alpha DVG Month	RETIRED AR...	2183184	1.0	Oracle's Alpha ...	DCP	Superceded
Patient Crack Posterior Tooth Therap...	RELEASED	2449302	1.0	the number of ...	NIDCR	
Post Brachytherapy Month Period Du...	APPRVD FOR ...	2201439	1.0	the post-brach...	CTEP	
Month Year Day	DRAFT NEW	2005378	3.0	PLEASE PROVI...	CTEP	
What month is it now?	RETIRED AR...	2003175	3.0	PLEASE PROVI...	CTEP	
day month year	RETIRED DEL...	2005190	3.0	PLEASE PROVI...	CTEP	

Figure 3.37 Search for Data Element dialog box with search results

If there are multiple pages of results, the **Previous** and **Next** buttons are enabled, allowing you to scroll through the results pages. In addition, you can resize both the search window and the columns within the results table, if necessary, to see more information in each field.

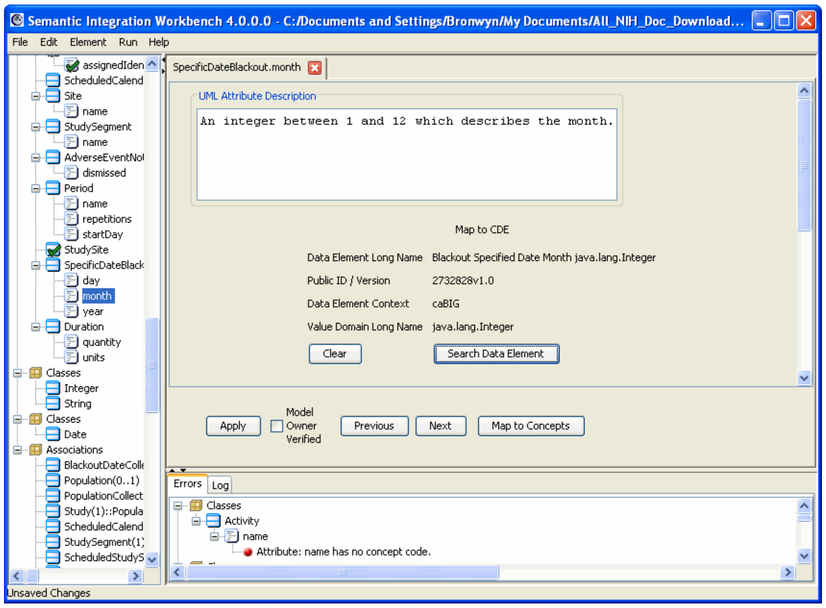
5. If the search returns no results (the Search dialog box remains empty), the element you want to map does not appear in the result set, or the search returns too many results, alter your search criteria and/or use the wildcard (*) to return different results. You may also choose to click **Close** to close the Search dialog box without mapping the attribute.
6. If the CDE you want to map appears in the result set, **double-click** anywhere in the row for the item. This closes the Search dialog box and enters the information for the selected CDE into the fields in the SIW viewer. It also maps (or verifies) the OC to the class associated with the attribute and the value domain to the datatype for the attribute.

Note: It is recommended that only data elements with a status of "Released" be used to map to model attributes.

You may receive an invalid selection error informing you that the DE you selected is mapped to a different OC than the current class element. This means that the selected element cannot be used for this mapping, and you must select a DE containing an OC that corresponds to the OC already mapped for the class element.

Another error you may encounter is that mapping the attribute to the selected DE causes a duplicate mapping. A duplicate mapping occurs when a different attribute in your model has already been mapped to the selected CDE.

If you select a valid CDE, the data element information appears in the detail view for the attribute.



An integer between 1 and 12 which describes the month.

Data Element Long Name: Blackout Specified Date Month java.lang.Integer
Public ID / Version: 2732828v1.0
Data Element Context: caBIG
Value Domain Long Name: java.lang.Integer

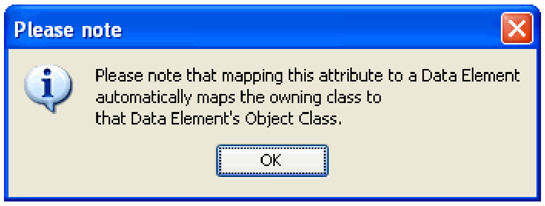
Attribute: name has no concept code.

Figure 3.38 Viewer showing an attribute mapped to a CDE (but not yet applied)

Once you have selected the CDE to map to the attribute, you have the following options:

- Click **Map to Concepts** to return to the concept information view. You may toggle between the concept and the CDE view for the attribute until the CDE mapping is applied. Once an attribute is mapped to a CDE, the concept information no longer appears.
- Click **Clear** to undo the choice of the CDE and then click **Apply**. This removes the CDE information mapping from the attribute.
- Click **Apply** to map the CDE to the attribute.

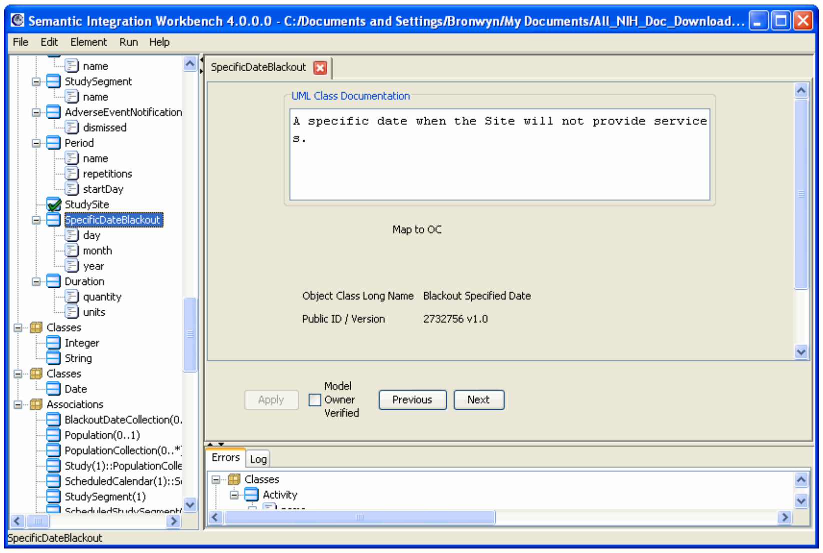
When an attribute is mapped to a CDE, the class that the attribute belongs to is mapped to the OC for the CDE selected, and the datatype for the attribute is mapped to the value domain for the CDE. If this is the first attribute for the class being mapped to a CDE, the SIW pops up a note informing you of the OC-to-class mapping.



Please note that mapping this attribute to a Data Element automatically maps the owning class to that Data Element's Object Class.

Figure 3.39 Pop-up dialog box informing you of the mapping of the class to the DE's Object Class

Like the CDE mapping for the attribute, the OC mapping for the class replaces the concept information displayed for the class.



A specific date when the Site will not provide services.

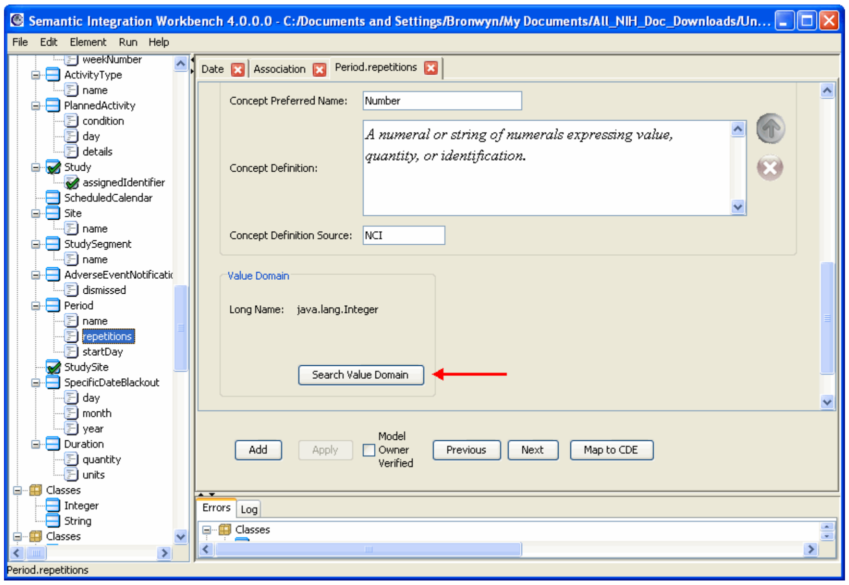
Object Class Long Name Blackout Specified Date
Public ID / Version 2732756 v1.0

Figure 3.40 Viewer showing a class element mapped to a caDSR Object Class

Mapping a UML Attribute to an Existing Value Domain

Mapping an attribute to a CDE automatically maps the datatype for the attribute to the value domain for the CDE. If an attribute is not mapped to a CDE, you can still map the attribute's datatype to a value domain from caDSR.

The option for mapping an attribute to a value domain appears below the concept information for the attribute in the detail view.



Concept Preferred Name: Number

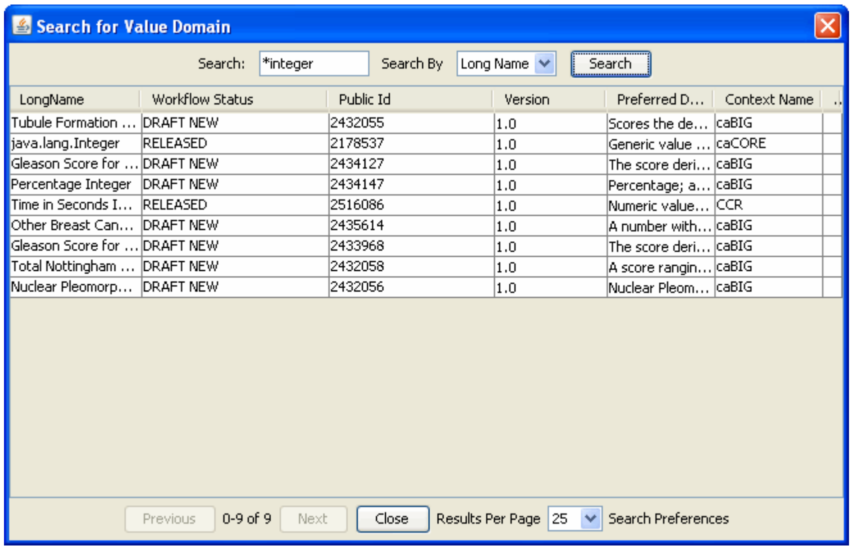
Concept Definition: *A numeral or string of numerals expressing value, quantity, or identification.*

Concept Definition Source: NCI

Long Name: java.lang.Integer

Figure 3.41 The Value Domain information box is located below the concept information

Clicking the **Search Value Domain** button opens the **Search for Value Domain** dialog box.



Search: *integer   Search By: Long Name   Results Per Page: 25

LongName	Workflow Status	Public Id	Version	Preferred D...	Context Name
Tubule Formation ...	DRAFT NEW	2432055	1.0	Scores the de...	caBIG
java.lang.Integer	RELEASED	2178537	1.0	Generic value ...	caCORE
Gleason Score for ...	DRAFT NEW	2434127	1.0	The score deri...	caBIG
Percentage Integer	DRAFT NEW	2434147	1.0	Percentage; a...	caBIG
Time in Seconds I...	RELEASED	2516086	1.0	Numeric value...	CCR
Other Breast Can...	DRAFT NEW	2435614	1.0	A number with...	caBIG
Gleason Score for ...	DRAFT NEW	2433968	1.0	The score deri...	caBIG
Total Nottingham ...	DRAFT NEW	2432058	1.0	A score rangin...	caBIG
Nuclear Pleomorp...	DRAFT NEW	2432056	1.0	Nuclear Pleom...	caBIG

Figure 3.42 Search for Value Domain from caDSR to map to the selected attribute

The Search for Value Domain dialog box functions similarly to other search functions in the SIW, allowing for the use of an asterisk (*) as a wildcard, and for resizing the dialog box and columns as necessary to better view the information.

By default, the search looks for a LongName that matches the information entered into the **Search** text box. You can also select to search by **Public ID**.

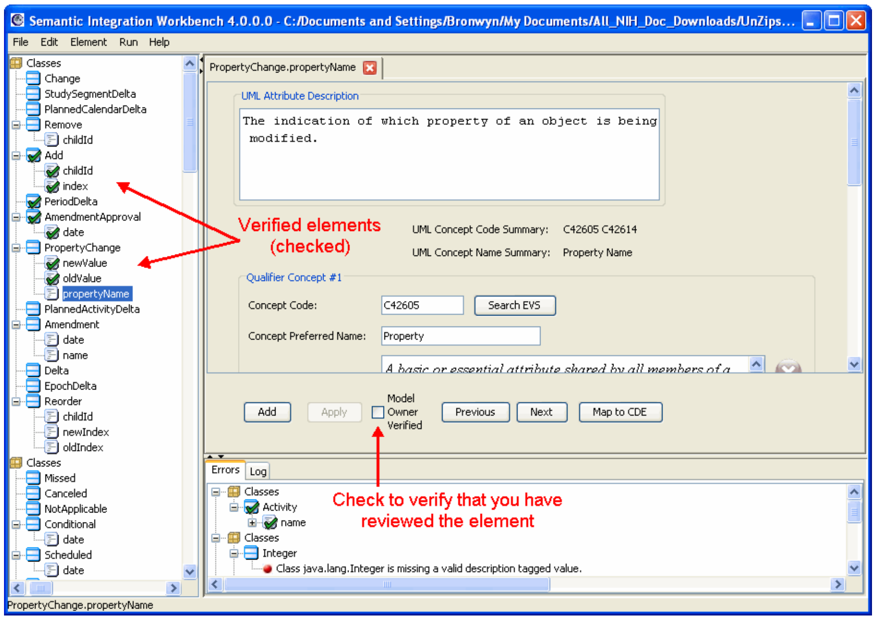
As with the other search functions, **double-clicking** an item in the result set maps the selected value domain to the selected attribute in the model.

After making changes to the value domain mapping, click **Apply** to apply those changes to the element.

Re-Verifying the Reviewed XMI File

Once you are finished reviewing the concept information in the file and where possible mapping elements to caDSR CDEs, and the XMI file is annotated to your satisfaction, you can use the **Model Owner Verified** checkbox to mark that the annotation for each element is complete.

Once an element has been checked as Model Owner Verified, a checkmark also appears next to the element in the navigation tree. This makes it very easy for you to identify which elements still need to be reviewed and verified.



The indication of which property of an object is being modified.

UML Concept Code Summary: C42605 C42614

UML Concept Name Summary: Property Name

Concept Code: C42605

Concept Preferred Name: Property

A basic or essential attribute shared by all members of a

Class java.lang.Integer is missing a valid description tagged value.

Figure 3.43 SIW Viewer in Review Annotated XMI File mode, with model owner verified elements

Once you are satisfied with the annotations for all of the elements in the file, you must mark all elements as **Model Owner Verified**. This includes the elements that you mapped to CDEs and those you simply verified the concept mapping for.

To verify an element:

1. Select an element from the navigation tree.
2. Review the information in the detail view to verify that the element is mapped to your satisfaction (either to one or more concepts or to an OC or CDE).
3. Click the **Model Owner Verified** checkbox to enable it. One of the following occurs, depending on the element you are verifying:
    - If you have selected a class element and there are un-verified attributes associated with that class, the next item in the tree is automatically selected for you to verify.

- If you have selected an attribute, a checkmark appears on the attribute in the navigation tree and the next item in the tree is automatically selected.
- If you have selected the last attribute in the class to be verified (all others have already been verified), and the class has already been verified, a checkmark appears on the attribute you are verifying and on the class element, and the next item in the tree is automatically selected.

The automatic movement to the next element during element verification is simply a navigational aid, so that you do not have to continually click **Next** to move to the next item. Once an element is verified, the SIW assumes that you are finished with that item.

Once you have reviewed all items in the file, and cleared all of the errors listed in the Errors tab (which may require multiple launches of the SIW and the XMI file to update the Errors tab), you must save the file using the **File > Save As** command. Reviewed and verified files should be saved with the term "approved" appended to the front of the filename. For example, if the original file name was `Annotated_FirstPass_myModel.xmi`, then you would save the reviewed file as `Approved_Annotated_FirstPass_myModel.xmi`. If the original file had a .uml extension (exported from ArgoUML) then the reviewed file must also be saved with a .uml extension.

For further instructions on saving a file, see *Saving Changes to a File* on page 39.

The reviewed approved annotated XMI file is used as input to the UML Loader by the NCICB staff. You can use the approved annotated XMI file as the input to the next version of your model, or to update the current version of the model with the SIW-generated annotations. See *Updating UML Model Definitions with SIW-Generated Changes* on page 40 for more information.

Troubleshooting

In addition to the troubleshooting tips listed below, the caDSR team maintains a list of commonly asked questions and tips about the SIW and UML Model Loader on the UML modeling FAQs wiki page located at: https://wiki.nci.nih.gov/x/7wFy.

The Status Bar

At the bottom of the main viewer window, the Status Bar displays confirmation and informative messages. Use it to confirm that a file was saved, that changes were applied or need to be applied.

Apply Changes vs. Saving Changes

Apply - Applies the changes made to the concept information. The Apply button is disabled until all concept fields are populated for newly added concepts, or until a change is made to existing concept information.

You must click **Apply** before navigating away from the changed item's node, or your changes will be discarded. The SIW prompts you to apply your changes if you attempt to view another node before clicking **Apply**.

Clicking **Apply** is not the same as saving the file. Apply simply applies your changes to the currently open session. You must still save the file (**File > Save** or **File > Save As**) to create an updated XMI file containing the applied changes.

Multiple Tabs Open

Each type of element in the SIW viewer will open in its own tab. this means that if you select different types of elements from the navigation tree to view, you will likely have multiple tabs open at once. Click the element title on each tab to see the information about that element; use the “X” on the tab to close the tab. See *Detail View* on page 29 for more information on tabbed viewing.

Search Dialog Box

The SIW contains several different search functions, including the EVS search, the data element search and the value domain search. The different search functionality for each item is similar but not identical. Furthermore, when you open a search dialog box to perform a search, you may find the previous search information still located in the dialog box and/or result set. Not all of the searches clear their information between search sessions.

All of the search features, however, allow you to resize both the dialog box and the column width of the result set, providing better visibility of the items listed. In addition, the **Previous** and **Next** buttons always function as “page turners” when there are multiple pages of results, and the **Close** button always closes the dialog box without performing any mapping.

For more information about the different search capabilities, see the following sections of this chapter:

- *Searching EVS for Concept Information* on page 62
- *Mapping a UML Attribute to an Existing Common Data Element* on page 83
- *Mapping a UML Attribute to an Existing Value Domain* on page 86.

Chapter 4
caCORE UML Loader and Registering Metadata

The UML Loader is a Java application that transforms UML domain models into caDSR metadata, reusing existing caDSR administered components or creating new ones as needed. This process is also referred to as registering the UML model in caDSR.

This chapter describes how the UML Loader transforms the UML model data from an annotated XMI file into caDSR metadata while also mapping the registered metadata back to the UML model that uses it.

Metadata registration into the caDSR is performed by a team at the NCICB, working with model owners to be sure that the information from their model is registered properly. The information in this chapter is designed to help model owners understand how their annotation work translates into caDSR content so that they can accurately predict the outcome of the loading of their model into caDSR.

Beyond the basic mapping of UML model elements to caDSR registered objects, this chapter also contains basic information about how association and inheritance relationships translate into caDSR content.

Topics in this chapter include:

- *Mapping Data Element Concepts* on page 107
- *Mapping Data Elements* on page 108
- *Mapping UML Associations to Object Class Relationships* on page 109
- *Mapping UML Inheritance* on page 110
- *Reviewing and Verifying Registered Metadata* on page 113

Note: Before continuing, you are encouraged to review *Terms You Should Know* on page 10 regarding the definitions of data elements and data element concepts, how they are formed, and how UML model elements relate to caDSR components.

Overview of UML Loader Process

Once the UML object model is annotated with EVS concepts and the annotated version of the model has been approved, the model owner sends the annotated XMI file to the NCICB caDSR team for processing. The EVS concept annotations contained in the file are the basis for determining whether each of the UML elements can be represented using existing caDSR components or if new ones must be created.

These EVS concept annotations function as the semantic bridge between the elements in the class model and the components in caDSR. This is why the UML Loader will not load XMI files that do not contain the mandatory EVS concept tags for all UML classes and attributes.

UML model element to caDSR component mapping is summarized in *Table 4.1*. Specifically, the UML Loader transforms UML model attributes and classes, including inheritance and association links, into caDSR metadata. UML class operations and other types of UML elements are *not* transformed. In addition, the UML Loader does not transform UML data models.

UML Model Element	***caDSR Administered Component***	***Additional Information***
Mapped EVS Concept (concept tagged value)	caDSR Concept (Concept Class). caDSR maintains its own concept data, corresponding to the information in EVS/NCI Thesaurus	The UML model concept information is correlated to caDSR concept information. If the model contains a concept not resident in caDSR, a new caDSR Concept is created for use.
Class	Object Class (OC)	Using the concept codes mapped to the class element, an existing caDSR OC is used or a new one is created.
Attribute	Property	Using the concept codes mapped to the class element, an existing caDSR Property is used or a new one is created.

Table 4.1 Transformation Mappings from UML Model Elements to caDSR Components

UML Model Element	***caDSR Administered Component***	***Additional Information***
Datatype (Attribute Datatype)	Value Domain)	This is the datatype associated with the attribute in the model. Typically the datatypes used are common Java datatypes and as such already exist in caDSR. However, some models define their own local datatypes (as classes in the model) and associate those with the attributes in the model using tagged values. Locally defined datatypes are referred to as “local value domains” or “LVD.”
Class-Attribute combination	Data Element Concept (DEC) A DEC is the combination of an OC and a Property.	After the OC and the Property are mapped (to the class and attribute), they are combined to form a DEC. If the combination already exists in caDSR the existing DEC is used. If the combination is new, a new DEC is created. One DEC is formed for every class-attribute combination in the model. So if the model has a class element “Gene” that has four attributes, four DECs are formed.
Class-Attribute-Datatype combination (The datatype is the datatype identified for the attribute.)	Data Element (DE) or Common Data Element (CDE).* A DE or CDE is the combination of a DEC with a value domain. *The terms DE and CDE are used interchangeably.	After the DEC is formed, the value domain is determined and added to the DEC to form the DE. If the combination already exists in caDSR the existing DE is used. If the combination is new, a new DE is created. One DE is formed for every class-attribute-datatype combination in the model. So if the model has a class element “Gene” with four attributes, there will be four DECs formed. Each of those DECs will be combined with the value domain corresponding to the datatype for each attribute, forming four DEs.

Table 4.1 Transformation Mappings from UML Model Elements to caDSR Components

So for every element in the UML model, a corresponding metadata component (also referred to as an administered component) in caDSR is either mapped or created (and then mapped). The “mapping” of caDSR components involves adding information about the project and the model to the component information in caDSR. This process of relating caDSR metadata to the models that use them is described in more detail in *Relating (Classifying) Metadata Items to UML Models* on page 94.

Submitting a UML Model to caDSR

When you have completed the semantic integration process and are ready to submit your UML model (via fully annotated and approved XMI file) to NCICB for registering, contact the NCICB help desk at: ncicb@pop.nci.nih.gov.

You will either be provided with the submission form and instructions, or pointed to the location of the form on GForge: http://gforge.nci.nih.gov/docman/index.php?group_id=64&selected_doc_group_id=184&language_id=1

You may want to follow the link to the submission form before you are ready to submit the file, in order to review the pre-requisite instructions included with the form.

Your model will initially be loaded to the caDSR Sandbox environment, allowing you to review the loading of your model and work with the caDSR curators to make any small changes needed. For more information on reviewing your model's elements after they are loaded, see *Reviewing and Verifying Registered Metadata* on page 113.

Relating (Classifying) Metadata Items to UML Models

Whenever caDSR metadata items are created or reused for a UML model, certain information about the model is appended to the item in the caDSR. This is done for two reasons:

1. It allows the information to be cross-referenced by Project Name and/or model information (class or attribute name);
2. It provides detailed information about how the Project/Model uses that particular caDSR component.

The relationship mapping is done by "Classifying" each component and by assigning an "Alternate Name" and "Alternate Definition" to each component.

The following table identifies the relationship between the information added to each component in caDSR and the UML model item that provides that information. Keep in mind that the more common caDSR components ("gene name" or "patient") is likely to have several of each of these associated with it, because each time a model uses the component, the relationship information is added to the component in caDSR.

caDSR Component Information	***UML Model Information***	***Used to...***
Classification Scheme (CS) Identifies the project in caDSR.	Project Name Provided by the model owner in the form submitted with the model to NCICIB.	All caDSR components derived from UML models are "classified" during model load. Classification identifies which projects use that caDSR component in their model. Classification also provides a method of categorization for caDSR items. You can search caDSR for a particular component or you can look for all components in a CS.

Table 4.2 Relationship between caDSR component information and UML model information.

caDSR Component Information	***UML Model Information***	***Used to...***
Classification Scheme Item (CSI) Identifies the sub-project, project alias or package name in caDSR	Sub-Project, Project Alias or Package Name Provided by the model owner or detected by the UML Loader (packages only).	The CSI is used as an additional classification or identification mechanism for when projects are divided into sub-projects or packages. Also like CS, CSIs provide an additional level of categorization for caDSR components.
Object Class Alternate Name	Class Name	Provides the exact name of the class element as used in the project's registered model, which has been mapped to this particular caDSR OC. The CS identifies the project.
Object Class Alternate Definition	Class description - Text is pulled from the description tagged value for the class element in the model.	Relates the identified project's intended use or definition of the class mapped to this particular caDSR OC. The CS identifies the project.
Data Element Alternate Name	Two are created: Class Name & Attribute Name (connected with a colon); Package Name & Attribute Name (referred to as a "fully qualified attribute.")	One provides the project's specific name for the class/attribute combination used to form this particular caDSR DE; the other provides the fully qualified attribute information for the attribute used to form the DE. The CS identifies the project; the fully qualified attribute name identifies the package or sub-project of the model containing the attribute.
Data Element Alternate Definition	Attribute description - Text is pulled from the description tagged value for the attribute element in the model.	Relates the project's intended use or definition for the attribute used to form this particular caDSR DE. The CS identifies the project.

Table 4.2 Relationship between caDSR component information and UML model information.

Note: Alternate Names and Alternate Definitions are also created for Property and Data Element Concept components, however these are not visible through the CDE Browser and they reiterate the Alternate Name/Definition information for Object Classes and Data Elements. They can be viewed through the CDE Admin Tool if necessary.

The sections that follow provide more detailed explanations for how and why caDSR metadata is classified, how and why alternate names and definitions are added to caDSR elements, and where you can view that information.

Classifying UML Model Metadata (Classification Schemes and Classification Scheme Items)

A *Classification* is a grouping of items related to a specific project or model. A *Classification Scheme* identifies the project and version associated with the registered model. A *Classification Scheme Item* identifies any sub-projects, packages, or aliases included in the project (since UML class models are commonly organized into packages). The UML Loader uses one classification scheme (CS) and at least one classification scheme item (CSI) for each UML domain model. Unless otherwise specified, all CS are created in the caBIG context.

The CS and CSI(s) for the domain model act as both a cross-referencing mechanism for viewing only a specific model and as an identifier for which projects are using what caDSR components in their models. Furthermore, classification schemes are used to identify whether or not that grouping of components is ready for general access through the CDE Browser. A CS will not appear in the CDE Browser until the workflow status is set to “Released.” This allows model owners and curators to review and edit their caDSR content before allowing full public visibility.

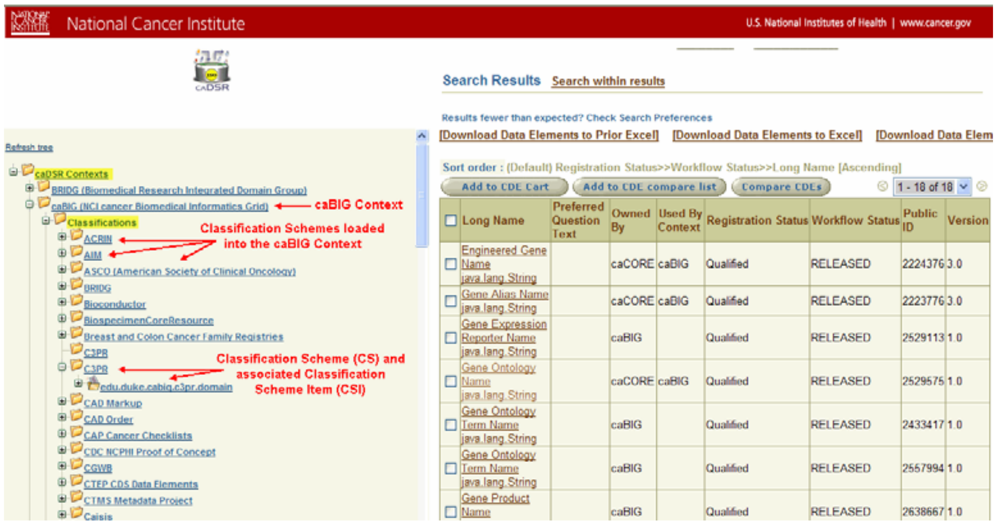
The CS for the project is created using the information provided on the form submitted with the XMI file to NCICB. This information includes Project Abbreviated Name, Project Long Name, Project Definition, and Version.

The CSI(s) for the project are created in one of two ways:

1. The UML Loader can be configured to create a CSI corresponding to each package in the UML model, using the package names from the XMI file and associating each CSI with the CS.
2. The UML Loader can be configured to ignore the separate packages in the UML model and instead have only one CSI specified for the entire model. The UML Loader then creates only one CSI associated with the CS.

When viewing caDSR components through the CDE Browser, the classification information can be viewed in two ways: in the tree view of the main CDE Browser page, and in the Classifications tab for a selected data element.

The CDE Browser tree view lists all Contexts that contain caDSR metadata. Expanding a Context (click the + sign to the left of the folder) displays the components of the context, including a node for classifications. Expanding the Classifications node provides a list of all classifications that have been loaded to that context.

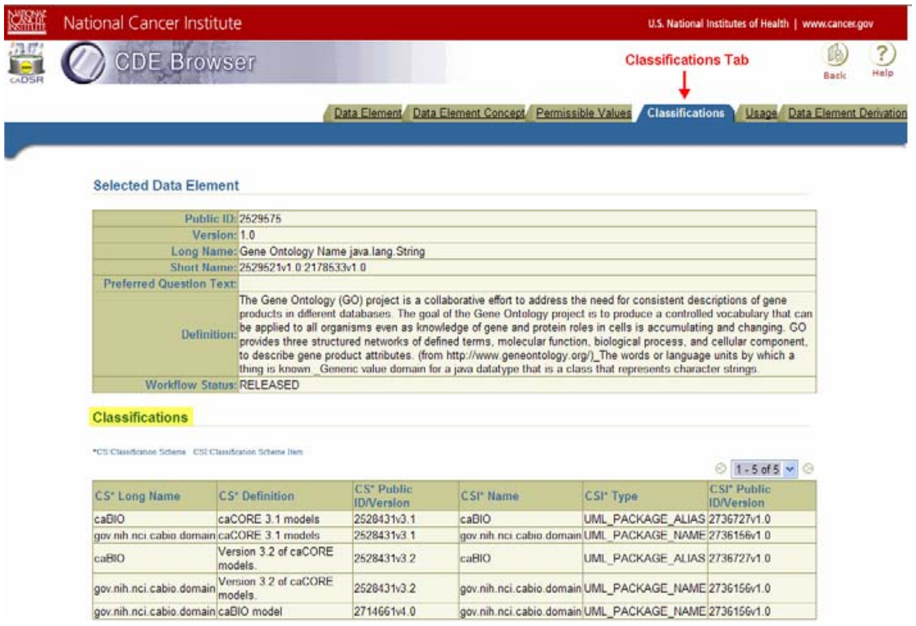

National Cancer Institute
U.S. National Institutes of Health | www.cancer.gov
caDSR
Refresh tree
caDSR Contexts
BRIDG (Biomedical Research Integrated Domain Group)
caBIG (NCI cancer Biomedical Informatics Grid)
caBIG Context
Classifications
Classification Schemes loaded into the caBIG Context
ACRIN
AIM
ASCO (American Society of Clinical Oncology)
BRIDG
Bioconductor
BiospecimenCoreResource
Breast and Colon Cancer Family Registries
C3PR
C3PR
Classification Scheme (CS) and associated Classification Scheme Item (CSI)
edu.duke.cabig.c3pr.domain
CAD Markup
CAD Order
CAP Cancer Checklists
CDC NCPHI Proof of Concept
CGWB
CTEP CDS Data Elements
CTMS Metadata Project
Caisis
Search Results
Search within results
Results fewer than expected? Check Search Preferences
[Download Data Elements to Prior Excel]
[Download Data Elements to Excel]
[Download Data Elem
Sort order : (Default) Registration Status>>Workflow Status>>Long Name [Ascending]
Add to CDE Cart
Add to CDE compare list
Compare CDEs
1 - 18 of 18

	Long Name	Preferred Question Text	Owned By	Used By Context	Registration Status	Workflow Status	Public ID	Version
☐	Engineered Gene Name java.lang.String		caCORE	caBIG	Qualified	RELEASED	2224376	3.0
☐	Gene Alias Name java.lang.String		caCORE	caBIG	Qualified	RELEASED	2223776	3.0
☐	Gene Expression Reporter Name java.lang.String		caBIG		Qualified	RELEASED	2529113	1.0
☐	Gene Ontology Name java.lang.String		caCORE	caBIG	Qualified	RELEASED	2529575	1.0
☐	Gene Ontology Term Name java.lang.String		caBIG		Qualified	RELEASED	2433417	1.0
☐	Gene Ontology Term Name java.lang.String		caBIG		Qualified	RELEASED	2557994	1.0
☐	Gene Product Name		caBIG		Qualified	RELEASED	2638667	1.0

Figure 4.1 CDE Browser tree-view with node items labeled

Selecting a CS or CSI in the tree-view allows you to search or browse for only the metadata elements used by that project/sub-project.

If you select to view a data element, one of the available tabs is called Classifications and lists all of the CS and CSI that are mapped to that data element.



Selected Data Element

Public ID:	2529575
Version:	1.0
Long Name:	Gene Ontology Name java.lang.String
Short Name:	2529521v1.0:2178533v1.0
Preferred Question Text:	
Definition:	The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene products in different databases. The goal of the Gene Ontology project is to produce a controlled vocabulary that can be applied to all organisms even as knowledge of gene and protein roles in cells is accumulating and changing. GO provides three structured networks of defined terms, molecular function, biological process, and cellular component, to describe gene product attributes. (from http://www.geneontology.org/)_The words or language units by which a thing is known._Generic value domain for a java datatype that is a class that represents character strings.
Workflow Status:	RELEASED

Classifications

*CS:Classification Scheme CSI:Classification Scheme Item

1 - 5 of 5

CS* Long Name	CS* Definition	CS* Public ID/Version	CSI* Name	CSI* Type	CSI* Public ID/Version
caBIO	caCORE 3.1 models	2528431v3.1	caBIO	UML_PACKAGE_ALIAS	2736727v1.0
gov.nih.nci.cabio.domain	caCORE 3.1 models	2528431v3.1	gov.nih.nci.cabio.domain	UML_PACKAGE_NAME	2736156v1.0
caBIO	Version 3.2 of caCORE models.	2528431v3.2	caBIO	UML_PACKAGE_ALIAS	2736727v1.0
gov.nih.nci.cabio.domain	Version 3.2 of caCORE models.	2528431v3.2	gov.nih.nci.cabio.domain	UML_PACKAGE_NAME	2736156v1.0
gov.nih.nci.cabio.domain	caBIO model	2714661v4.0	gov.nih.nci.cabio.domain	UML_PACKAGE_NAME	2736156v1.0

Figure 4.2 CDE Browser showing the Classification Tab for a Data Element

The classifications listing shows every CSI that uses the selected data element. This means that the CS may be listed several times, once for each CSI it contains that uses that DE.

CS and CSI information is also listed with each Alternate Name and Definition in the main Data Element tab. These are discussed in more detail in the section below.

Alternate Names and Alternate Definitions

Alternate Names and Alternate Definitions are used to provide specific meaning for how or why a particular model used a particular piece of metadata. This is done by providing both the exact name of the element(s) involved (class name and/or attribute name) and the element description resident in the model.

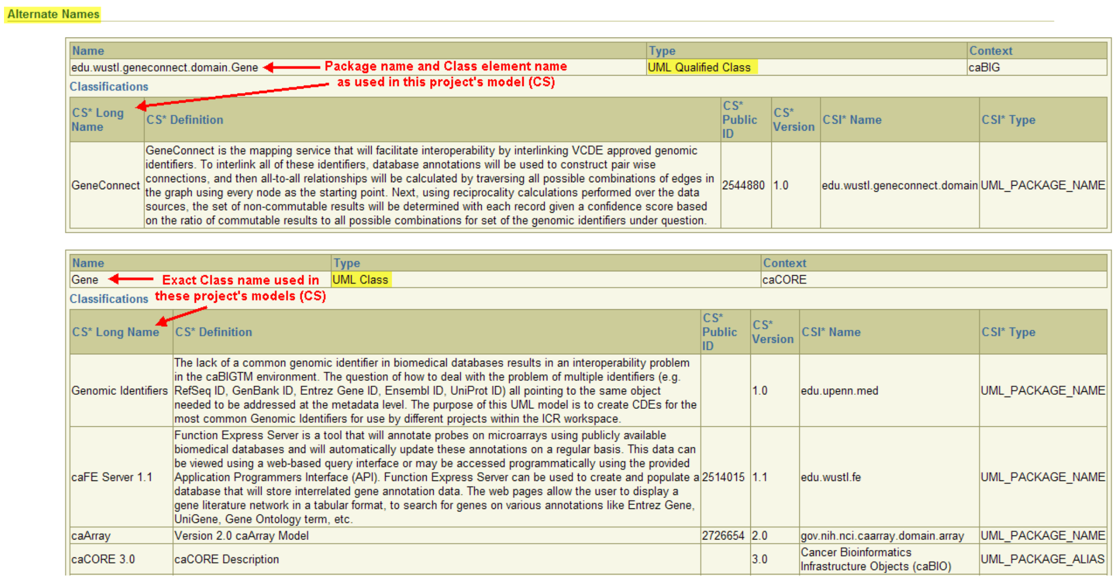
Because this information is specific to a particular project's model, Alternate Names and Definitions are always listed with the CS and CSI for the model.

Alternate Names and Alternate Definitions are created for all caDSR components, however the CDE Browser only displays this information for Object Classes and for data elements. The caDSR Admin Tool shows all associated alternate information.

Creating Alternate Names and Alternate Definitions for Object Classes

Each time a UML model class element is mapped to a caDSR OC, two Alternate Names are created for that OC:

- A **UML Class** type Alternate Name: Provides the exact class name of the element in the model. The project is identified by the CS/CSI listed with the alternate name. The **Type** column shows **UML Class**.
- A **UML Qualified Class** type Alternate Name: Provides the exact package name and class name combination for the class element. The project is identified by the CS/CSI listed with the alternate name. The **Type** column shows **UML Qualified Class**. For example, the class name "Gene" in the domain "gov.nih.nci.cabio.domain" would become an Alternate Name of **gov.nih.nci.cabio.domain.Gene**" with a Type of **UML Qualified Class**.


Alternate Names

Name	Type	Context
edu.wustl.geneconnect.domain.Gene	UML Qualified Class	caBIG

Package name and Class element name
as used in this project's model (CS)

Classifications

CS* Long Name	CS* Definition	CS* Public ID	CS* Version	CSI* Name	CSI* Type
GeneConnect	GeneConnect is the mapping service that will facilitate interoperability by interlinking VCDE approved genomic identifiers. To interlink all of these identifiers, database annotations will be used to construct pair wise connections, and then all-to-all relationships will be calculated by traversing all possible combinations of edges in the graph using every node as the starting point. Next, using reciprocality calculations performed over the data sources, the set of non-commutable results will be determined with each record given a confidence score based on the ratio of commutable results to all possible combinations for set of the genomic identifiers under question.	2544880	1.0	edu.wustl.geneconnect.domain	UML_PACKAGE_NAME

Name	Type	Context
Gene	UML Class	caCORE

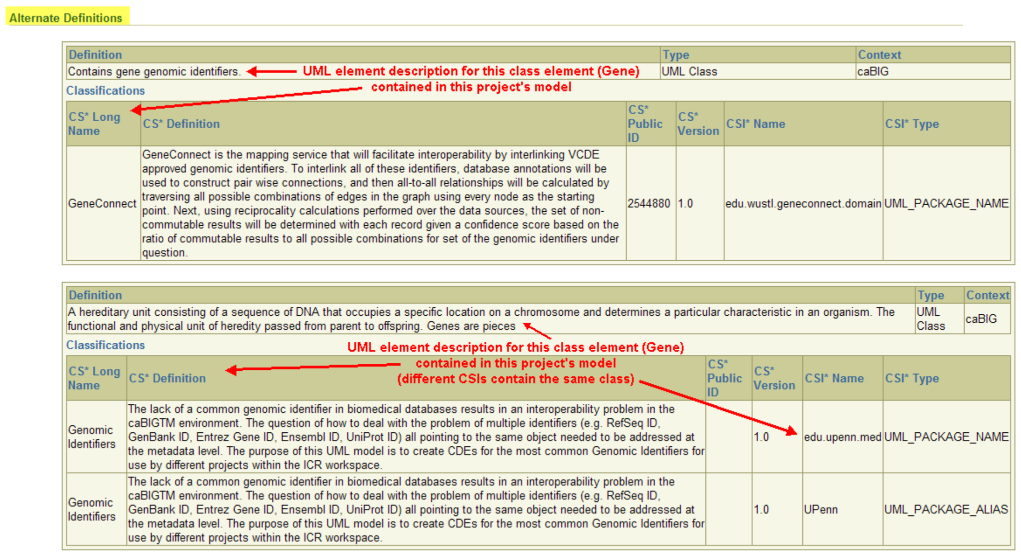
Classifications

CS* Long Name	CS* Definition	CS* Public ID	CS* Version	CSI* Name	CSI* Type
Genomic Identifiers	The lack of a common genomic identifier in biomedical databases results in an interoperability problem in the caBIGTM environment. The question of how to deal with the problem of multiple identifiers (e.g. RefSeq ID, GenBank ID, Entrez Gene ID, Ensembl ID, UniProt ID) all pointing to the same object needed to be addressed at the metadata level. The purpose of this UML model is to create CDEs for the most common Genomic Identifiers for use by different projects within the ICR workspace.		1.0	edu.upenn.med	UML_PACKAGE_NAME
caFE Server 1.1	Function Express Server is a tool that will annotate probes on microarrays using publicly available biomedical databases and will automatically update these annotations on a regular basis. This data can be viewed using a web-based query interface or may be accessed programmatically using the provided Application Programmers Interface (API). Function Express Server can be used to create and populate a database that will store interrelated gene annotation data. The web pages allow the user to display a gene literature network in a tabular format, to search for genes on various annotations like Entrez Gene, UniGene, Gene Ontology term, etc.	2514015	1.1	edu.wustl.fe	UML_PACKAGE_NAME
caArray	Version 2.0 caArray Model	2726654	2.0	gov.nih.nci.caarray.domain.array	UML_PACKAGE_NAME
caCORE 3.0	caCORE Description		3.0	Cancer Bioinformatics Infrastructure Objects (caBIO)	UML_PACKAGE_ALIAS

Figure 4.3 CDE Browser - Object Class View showing Alternate Names for the OC.

For common classes, such as Gene, many different projects are likely to have used the same class name for the item. In this case, the UML Class type Alternate Name will list all projects, packages, and versions where a class element with that name exists.

A single Alternate Definition is created for each class element mapped to a caDSR OC. This definition is given the type **UML Class** and comes from the element description tagged value(s) associated with the class element in the model. As with all alternate information, the project is identified by the CS/CSI listed with the definition.


UML element description for this class element (Gene) contained in this project's model
UML element description for this class element (Gene) contained in this project's model (different CSIs contain the same class)

Alternate Definitions

Definition	Type	Context
Contains gene genomic identifiers.	UML Class	caBIG

Classifications

CS* Long Name	CS* Definition	CS* Public ID	CS* Version	CSI* Name	CSI* Type
GeneConnect	GeneConnect is the mapping service that will facilitate interoperability by interlinking VCDE approved genomic identifiers. To interlink all of these identifiers, database annotations will be used to construct pair wise connections, and then all-to-all relationships will be calculated by traversing all possible combinations of edges in the graph using every node as the starting point. Next, using reciprocality calculations performed over the data sources, the set of non-commutable results will be determined with each record given a confidence score based on the ratio of commutable results to all possible combinations for set of the genomic identifiers under question.	2544880	1.0	edu.wustl.geneconnect.domain	UML_PACKAGE_NAME

Definition	Type	Context
A hereditary unit consisting of a sequence of DNA that occupies a specific location on a chromosome and determines a particular characteristic in an organism. The functional and physical unit of heredity passed from parent to offspring. Genes are pieces	UML Class	caBIG

Classifications

CS* Long Name	CS* Definition	CS* Public ID	CS* Version	CSI* Name	CSI* Type
Genomic Identifiers	The lack of a common genomic identifier in biomedical databases results in an interoperability problem in the caBIGTM environment. The question of how to deal with the problem of multiple identifiers (e.g. RefSeq ID, GenBank ID, Entrez Gene ID, Ensembl ID, UniProt ID) all pointing to the same object needed to be addressed at the metadata level. The purpose of this UML model is to create CDEs for the most common Genomic Identifiers for use by different projects within the ICR workspace.		1.0	edu.upenn.med	UML_PACKAGE_NAME
Genomic Identifiers	The lack of a common genomic identifier in biomedical databases results in an interoperability problem in the caBIGTM environment. The question of how to deal with the problem of multiple identifiers (e.g. RefSeq ID, GenBank ID, Entrez Gene ID, Ensembl ID, UniProt ID) all pointing to the same object needed to be addressed at the metadata level. The purpose of this UML model is to create CDEs for the most common Genomic Identifiers for use by different projects within the ICR workspace.		1.0	UPenn	UML_PACKAGE_ALIAS

Figure 4.4 CDE Browser - Object Class View showing Alternate Definitions for the OC.

Many Alternate Definitions will have only one CS/CSI listed, while others will have several. If the class is used either by multiple versions of a project and/or by multiple packages within a model, each of those uses is listed because the class element is defined the same way throughout the model; all of the CS/CSI listed use the exact same element description within the model.

Creating an Alternate Names and Alternate Definitions for Data Elements

Alternate Names and Alternate Definitions for data elements are created using a combination of the class and attribute element details from each model.

Each time UML model elements are mapped to form a caDSR DE, two Alternate Names are created for that DE:

- A **UML Class:UML Attr** type Alternate Name: Provides the exact class/attribute name combination of the mapped elements in the model, separated by a colon. The **Type** column shows **UML Class:UML Attr**.

  For example, the class element “Gene” having an attribute “name” would become an Alternate Name of **Gene:name** with a Type of **UML Class:UML Attribute**.

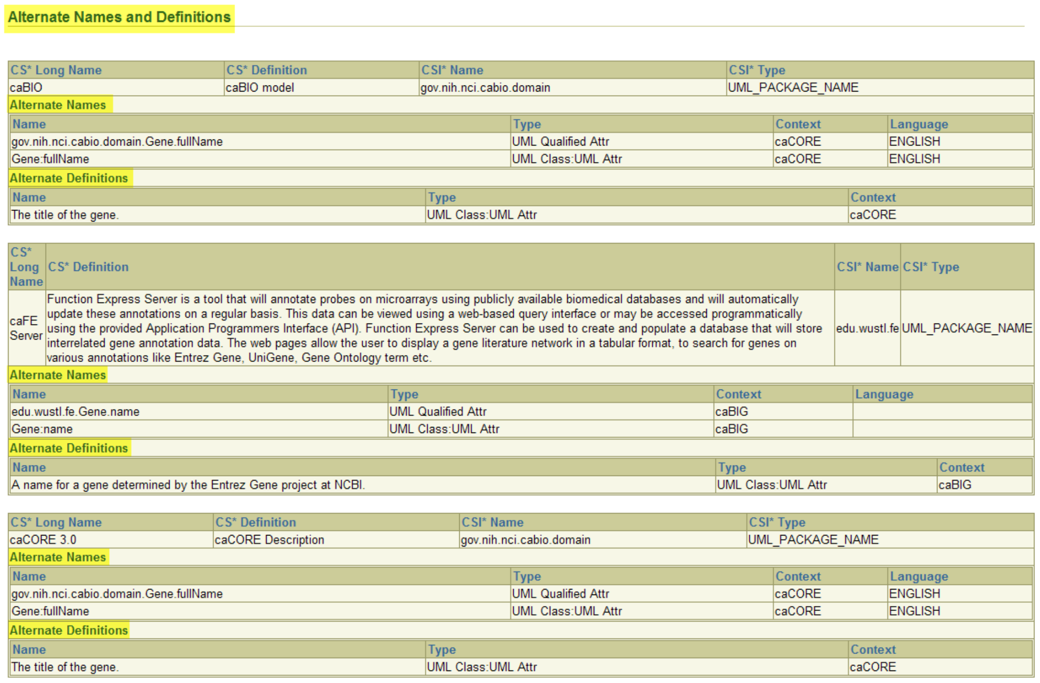
- A **UML Qualified Attr** type Alternate Name: Provides the exact package name, class name and attribute name combination for the mapped elements in the model, separated by periods. The **Type** column shows **UML Qualified Attr**.

  For example, the class element “Gene” having an attribute “name” in the domain “gov.nih.nci.cabio.domain” would become an Alternate Name of **gov.nih.nci.cabio.domain.Gene.name**” with a Type of **UML Qualified Attr**.

A single Alternate Definition is created for each class/attribute combination mapped to a caDSR DE. This definition is given the type **UML Class:UML Attr** and comes from the element description tagged value(s) associated with the attribute element in the model.

The UML attribute description was chosen as opposed to the combination of the description of the class and attribute because, in practice, when defining an attribute for a class, the notion of the class is generally incorporated in the attribute's description.

For DEs each CS/CSI is listed with the associated alternate information, as opposed to listing the Alternate Names and Alternate Definitions separately as is the case for OCs.


Alternate Names and Definitions

CS* Long Name	CS* Definition	CSI* Name	CSI* Type
caBIO	caBIO model	gov.nih.nci.cabio.domain	UML_PACKAGE_NAME

Alternate Names

Name	Type	Context	Language
gov.nih.nci.cabio.domain.Gene.fullName	UML Qualified Attr	caCORE	ENGLISH
Gene:fullName	UML Class:UML Attr	caCORE	ENGLISH

Alternate Definitions

Name	Type	Context
The title of the gene.	UML Class:UML Attr	caCORE

CS* Long Name	CS* Definition	CSI* Name	CSI* Type
caFE Server	Function Express Server is a tool that will annotate probes on microarrays using publicly available biomedical databases and will automatically update these annotations on a regular basis. This data can be viewed using a web-based query interface or may be accessed programmatically using the provided Application Programmers Interface (API). Function Express Server can be used to create and populate a database that will store interrelated gene annotation data. The web pages allow the user to display a gene literature network in a tabular format, to search for genes on various annotations like Entrez Gene, UniGene, Gene Ontology term etc.	edu.wustl.fe	UML_PACKAGE_NAME

Alternate Names

Name	Type	Context	Language
edu.wustl.fe.Gene.name	UML Qualified Attr	caBIG	
Gene:name	UML Class:UML Attr	caBIG	

Alternate Definitions

Name	Type	Context
A name for a gene determined by the Entrez Gene project at NCBI.	UML Class:UML Attr	caBIG

CS* Long Name	CS* Definition	CSI* Name	CSI* Type
caCORE 3.0	caCORE Description	gov.nih.nci.cabio.domain	UML_PACKAGE_NAME

Alternate Names

Name	Type	Context	Language
gov.nih.nci.cabio.domain.Gene.fullName	UML Qualified Attr	caCORE	ENGLISH
Gene:fullName	UML Class:UML Attr	caCORE	ENGLISH

Alternate Definitions

Name	Type	Context
The title of the gene.	UML Class:UML Attr	caCORE

Figure 4.5 CDE Browser - Data Element view showing Alternate Names and Definitions

Creating Alternate Names and Alternate Definitions for a Property

Because the caDSR Property metadata represents the attribute element in a UML model, the Alternate Names and Alternate Definitions for Properties are also based on the attribute information provided in the model.

A single Alternate Name is created for each attribute element mapped to a caDSR Property. This name is given the type **UML Attribute** and comes from the exact name used for the attribute in the model. A single Alternate Definition is also created for the attribute element mapped to the Property. This definition is given the type **UML Attribute** and comes from the element description tagged value(s) associated with the attribute element in the class model.

While alternate information for caDSR Properties is not viewable in the CDE Browser, it does appear in the caDSR Admin Tool.

Creating Alternate Names and Alternate Definitions for Data Element Concepts

Because caDSR DECs the class/attribute element combination in a UML model, the Alternate Names and Alternate Definitions for DECs are also based on a combination of the class/attribute information provided in the model.

A single Alternate Name is created for each class/attribute combination element associated with a caDSR DEC. This name is given the type **UML Class:Attribute** and comes from the exact names used for the elements in the model. A single Alternate Definition is also created for the class/attribute combination mapped to the DEC. This definition is also given the type **UML Class:Attribute** and comes from the element description tagged value(s) associated with both the class and attribute elements in the class model.

While alternate information for DECs is not viewable in the CDE Browser, it does appear in the caDSR Admin Tool.

Using Pre-mapped CDEs (Common Data Elements)

The SIW interface provides the ability to map UML model elements to existing caDSR CDEs if appropriate. Specifically, it is the attribute elements in the model that are mapped to CDEs. When this is done, the class element is automatically mapped to the OC for the CDE, and the datatype is mapped to the value domain for the CDE.

Because some of this mapping can be pre-entered, the UML Loader automatically checks each attribute in the model to see if it contains the following tagged values:

- CADSR_DE_ID
- CADSR_DE_VERSION

If both of these tagged values are present, the DE (CDE) and all of its composite parts (the OC, Property, value domain, and DEC) are reused and the caDSR components are classified using the CS/CSI information provided for the project/model.

In addition, if the context that owns the mapped CDE is different than the context to which the model is being loaded, the Alternate Name type shown for the component lists “Used_By” and no Alternate Definition appears. You must view the CDE within the context that owns it to see this information.

Creating and Updating Concepts in caDSR

Because trying to retrieve concept information directly from EVS could make the UML Loader cumbersome and slow, and because new concept information is often introduced well before EVS can be updated, the caDSR maintains its own repository of concept information that corresponds to the NCI Thesaurus information in EVS.

When the UML Loader begins to process a file, it checks to see if each concept corresponding to the specified concept code already exists in caDSR. If the concept code exists but the name or definition is different, the UML Loader first checks EVS to see if the name and definition match what is in the XMI file. If they are, the loader updates the concept in caDSR with the new information.

If the code does not exist, then a new concept is created in caDSR. The data used for creating the new concept is pulled from the tagged values located in the XMI file which were provided through the annotation process. The specific mapping is shown in *Table 4.3*. using the example of the concept for Gene.

Concept Attribute	***Data***	***Example***
Preferred Name	Derived from ConceptCode tagged value.	C16612
Long Name	Derived from ConceptPreferredName tagged value.	Gene
Preferred Definition	Derived from ConceptDefinition tagged value.	A functional unit of heredity which occupies a specific position (locus) on a particular chromosome, is capable of reproducing itself exactly at each cell division, and directs the formation of a protein or other product.
Version	1.0 (Default)	1.0
Workflow Status	RELEASED (Default)	Released
Context	Specified in the submission form with the model - the context to which the model is to be uploaded.	caBIG
Begin Date	Current Timestamp	01/23/2005

Table 4.3 Concept Attribute details

Definitions that come from sources other than NCI are captured as Alternate Definitions for a concept and are mapped as shown in *Table 4.4*.

Alternate Definition	***Data***
Definition	Derived from ConceptDefinition tagged value.
Context	Specified in the submission form with the model - the context to which the model is to be uploaded.

Table 4.4 Alternate Definition details

Mapping a UML Class to an Object Class

Each class in the UML domain model is mapped to a caDSR Object Class (OC). The UML Loader performs this mapping based on the NCI concepts to which the UML class elements have been mapped through the SIW. These are included as tagged values in the annotated XMI file for the UML model.

To map a UML class to a caDSR Object Class, the UML Loader retrieves the NCI concept codes for the UML class from the tagged values in the XMI file and checks to see if a caDSR Object Class based on those values exists. If it exists, the UML model class is mapped to it, and the OC is classified and appended with the appropriate CS/CSI and alternate name/definition information as described in *Relating (Classifying) Metadata Items to UML Models* on page 94.

If there is no OC in caDSR based on the mapped NCI concept code values for the class, the UML Loader must create a new OC. *Table 4.5* illustrates the details of a new object class that UML Loader creates.

Object Class Attribute	***Description***	***Example***
Preferred Name	Derived from the concept codes of the underlying concept(s). Usually the concept identifier(s).	C40992
Long Name	Derived from the long name of underlying concept(s).	Homologous Protein
Preferred Definition	Derived from the preferred definition of underlying concept(s).	A protein similar in structure and evolutionary origin to a protein in another species.
Version	1.0 (default)	1.0
Workflow Status	RELEASED (default)	Released
Context	Specified in the submission form with the model - the context to which the model is to be uploaded. Default is caBIG.	caBIG
Begin Date	Current Timestamp	01/23/2005

Table 4.5 Object class attribute details

As with all caDSR components, the OC is classified and appended with the appropriate CS/CSI and alternate name/definition information from the UML model as described in *Relating (Classifying) Metadata Items to UML Models* on page 94.

Mapping a UML Attribute to a Property

Each attribute in the UML domain model is mapped to a caDSR Property. The UML Loader performs this mapping based on the NCI concepts to which the UML attribute elements have been mapped through the SIW. These are included as tagged values in the annotated XMI file for the UML model.

To map a UML attribute to a caDSR Property, the UML Loader retrieves the NCI concept codes for the UML attribute from the tagged values in the XMI file and checks to see if a caDSR Property based on those values exists. If it exists, the UML model attribute is mapped to it, and the Property is classified and appended with the appropriate CS/CSI and alternate name/definition information as described in *Relating (Classifying) Metadata Items to UML Models* on page 94.

If there is no Property in caDSR based on the mapped NCI concept code values for the attribute, the UML Loader must create a new one. *Table 4.6* illustrates the details of the new Property that the UML Loader creates.

Property Attribute	***Description***	***Example***
Preferred Name	Derived from the concept codes of the underlying concept(s). Usually the concept identifier(s).	C25552:C411167
Long Name	Derived from the long name of underlying concept(s).	Lead:Organization alUnit
Preferred Definition	Derived from the preferred definition of underlying concept(s).	Be in charge of.: Organizational unit like a laboratory, institute or consortium.
Version	1.0 (default)	1.0
Workflow Status	RELEASED (default)	Released
Context	Specified in the submission form with the model - the context to which the model is to be uploaded. Default is caBIG.	caBIG
Begin Date	Current Timestamp	01/23/2005

Table 4.6 Property attribute details

Mapping a UML Class to a Value Domain

Each class stereotyped as “CADSR Value Domain” or “enumeration” in the UML domain model is mapped to a caDSR value domain. The UML Loader performs this mapping using the Long Name of the value domain and the Context to which it is being loaded.

For each class with Stereotype “CADSR Value Domain,” the UML Loader looks for the following tagged Values:

- CADSR_ValueDomainDefinition
- CADSR_ValueDomainDatatype
- CADSR_ValueDomainType (E or N)
- CADSR_ConceptualDomainPublicID
- CADSR_ConceptualDomainVersion
- CADSR_RepresentationPublicID
- CADSR_RepresentationVersion

and optionally:

- ValueDomainConceptCode
- ValueDomainConceptPreferedName
- ValueDomainConceptDefinition[_n]

- ValueDomainConceptDefinitionSource
- Value DomainQualifierConceptCodeN
- ValueDomainQualifierConceptPreferredNameN
- ValueDomainQualifierConceptDefinitionN[_n]
- ValueDomainQualifierConceptDefinitionSourceN

The list of optional tags above are used to indicate a “top level” concept for a value domain. This is also referred to as a “referenced value domain.”

If the value domain is non-enumerated (N), the reference to a top level concept indicates that the values are not explicitly listed as attributes of the value domain class, but that the data values for the attribute are constrained to children of the top level concept.

If the value domain is enumerated (E) the top level concept indicates that the permitted data values are listed explicitly as attributes of the value domain class and are children of the top level concept.

During the load process, the UML Loader starts by creating a value domain for each class with Stereotype “CADSR Value Domain.” *Table 4.7* below lists the information used to create each value domain.

caDSR Field	***Value Used***
Long Name	UML Class Name
Preferred Name	Similar to the generated public ID and version
Preferred Definition	From 'ValueDomainDefinition' tagged value
Value Domain Type	From 'ValueDomainType' tagged value
Context	From run time default values provided by model owner
Version	1.0
Workflow Status	Specified as Default (e.g DRAFT NEW)
Conceptual Domain	From 'ConceptualDomainPublicID' and 'ConceptualDomainVersion' tagged values
Datatype	From 'ValueDomainDatatype'
Begin Date	The date of the load
Concept Derivation Rule	Optionally created from the list of Concepts in tagged values. Similar to Object Class Concept Derivation Rules

Table 4.7 caDSR fields for creating a Value Domain

Value Meanings

For each attribute associated with a “CADSR Value Domain” stereotyped class, the UML Loader looks for the following tagged values:

- ValueMeaningConceptCode
- ValueMeaningConceptPreferredName

- ValueMeaningConceptDefinition[_n]
- ValueMeaningConceptDefinitionSource
- ValueMeaningQualifierConceptCodeN
- ValueMeaningQualifierConceptPreferredNameN
- ValueMeaningQualifierConceptDefinitionN[_n]
- ValueMeaningQualifierConceptDefinitionSourceN

During the load process and after each value domain has been loaded, the UML Loader looks for a matching caDSR Value Meaning based on the Value Meaning name or the mapped concepts, if present. If an existing Value Meaning is found, the UML Loader reuses it. If no match is found, the UML Loader creates a new Value Meaning. *Table 4.8* below lists the information used to create each Value Meaning.

caDSR Field	***Value Used***
Value Meaning	Created from the concatenation of 'ValueMeaningConceptPreferredName' and qualifiers if present.
Value Meaning Description	Concatenation of 'ValueMeaningConceptDefinition[_n]' tagged values and qualifiers if present.
Begin Date	The date of the load
Concept Derivation Rule	Created from the list of Concept in tagged values. Similar to Object Class and Properties Concept Derivation Rules

Table 4.8 caDSR fields used for Value Meaning

Permissible Values

For each Value Meaning, UML Loader creates a new permissible value. *Table 4.9* below lists the information used to create these values.

caDSR Field	***Value Used***
Value	From the underlying attribute's name
Value Meaning	The existing Value Meaning name, or the Value Meaning created according to *Table 4.8*.
Meaning Description	the Value Meaning Description associated with the existing Value Meaning, or the one created according to*Table 4.8*
Begin Date	The date of the load

Table 4.9 caDSR fields used for Permissible Values

Using a Value Domain Defined within the Model

In order for an attribute in the UML model to use a value domain defined within the model (also referred to as a *local value domain* or LVD), the attribute must already have a tagged value of type “CADSR Local Value Domain” associated with it. The value indicated in the tag is the name given to the LVD.

For example: The contains a class that is stereotyped as “CADSR Value Domain” with the name “My Model Value Domain.” For a UML attribute to use this value domain, attribute must have a tagged value of: “CADSR Local Value Domain” / “My Model Value Domain” (name/value).

Mapping Data Element Concepts

A Data Element Concept (DEC) is formed using the combination of a caDSR Object Class and a caDSR Property. Each class/attribute combination in the UML domain model is mapped to a caDSR DEC. The UML Loader performs this mapping based on the mapping it has already done for the model classes (to caDSR OCs) and the model attributes (caDSR Properties) as described in the previous sections.

The UML Loader checks to see if a caDSR DEC based on the OC and Property values for the class/attribute combination exists. If it exists, the UML model class/attribute combination is mapped to it, and the DEC is classified and appended with the appropriate CS/CSI and alternate name/definition information as described in *Relating (Classifying) Metadata Items to UML Models* on page 94.

If there is no DEC in caDSR based on the OC and Property values for the class/attribute combination, the UML Loader must create a new one. *Table 4.10* illustrates the details of the new DEC that the UML Loader creates.

Data Element Concept Attribute	***Description***	***Example***
Preferred Name	Derived from the Object Class Public ID and version and Property Public ID and version. A colon is used as the separator character between these values.	1111111v1.0:2222222v1.0
Long Name	Derived from the Object Class Long Name and Property Long Name. A space is used as the separator character between these two values.	Homologous Protein Alignment Length
Preferred Definition	Object Class Preferred Definition and Property Preferred Definition. A colon is used as the separator character between these two values.	A protein similar in structure and evolutionary origin to a protein in another species: The linear extent in space from one end to the other. Often used synonymously with distance.
Version	1.0 (default)	3.0
Workflow Status	RELEASED (default)	Draft New
Context	Specified in the submission form with the model - the context to which the model is to be uploaded. Default is caBIG.	caBIG
Begin Date	Current Timestamp	01/23/2005

Table 4.10 Data Element Concept details

Data Element Concept Attribute	***Description***	***Example***
Object Class Long Name	Object Class corresponding to the UML Class	Homologous Protein
Property Long Name	Property corresponding to the UML Attribute	Alignment Length

Table 4.10 Data Element Concept details

Mapping Data Elements

A Data Element (DE or CDE) is formed using the combination of a caDSR DEC and a value domain. Each class/attribute/datatype combination in the UML model is mapped to a caDSR DE. The UML Loader performs this mapping based on the mapping it has already done for the class/attribute combinations in the model (to caDSR DECs) and a generic existing caDSR value domain that corresponds to the datatype of the attribute or a value domain defined in the model.

The UML Loader checks to see if a caDSR DE based on the DEC and value domain combination exists. If it exists, the UML model class/attribute/datatype combination is mapped to it, and the DE is classified and appended with the appropriate CS/CSI and alternate name/definition information as described in *Relating (Classifying) Metadata Items to UML Models* on page 94.

Note: If the Context into which the UML model is being loaded is different from the Context that owns the existing data element, a "USED_BY" designation is added to the existing data element to capture the use by another Context

If there is no DE in caDSR based on the DEC and value domain combination, the UML Loader must create a new one. *Table 4.11* illustrates the details of the new DE that the UML Loader creates.

Data Element Attribute	***Description***	***Example***
Preferred Name	Derived from the DEC Public ID and version and the Value Domain Public ID and version. A colon is the separator character.	3333333v1.0v:4444444v 1.0
Long Name	Derived from the DEC Long Name and Value Domain Long Name. A space is the separator character.	Homologous Protein Alignment Length java.lang.Long

Table 4.11 Data Element details

Data Element Attribute	***Description***	***Example***
Preferred Definition	Derived from the DEC Preferred Definition and Value Domain Preferred Definition. The separator character is a colon.	A protein similar in structure and evolutionary origin to a protein in another species: The linear extent in space from one end to the other. Often used synonymously with distance. Value Domain for java language ‘java.lang.Long’ datatype.
Version	1.0 (default)	3.2
Workflow Status	RELEASED (default)	Draft New
Context	Specified in the submission form with the model - the context to which the model is to be uploaded. Default is caBIG.	caBIG
Begin Date	Current Timestamp	01/24/2005
Data Element Concept Long Name	The Data Element Concept corresponding to the UML Class and one of its attributes.	Homologous Protein Alignment Length
Value Domain	Corresponds to the datatype of the attribute. Supported datatypes include java classes and java primitives. **Note:** An updated file that defines the mapping between datatypes and value domains is located here: http://cadsrsiw.nci.nih.gov/datatype-mapping.xml.	java.lang.Long

Table 4.11 Data Element details (Continued)

Mapping UML Associations to Object Class Relationships

Each association in the UML domain model is mapped to an Object Class Relationship in caDSR.

Table 4.12 below provides details regarding the information used by the UML Loader to create new Object Class Relationships.

Object Class Relationship Attribute	***Data***
Preferred Name	Generated, equals source class corresponding Object Class public ID + version; target class corresponding Object Class and version.
Long Name	Derived from the role name of underlying association.

Table 4.12 New Object Class Relationship details

Object Class Relationship Attribute	***Data***
Preferred Definition	Derived from the type of association. Example of the derived value for preferred definition: Zero-to-Many Zero-to-One Many-to-One One-to-Many Many-to-Many Generalizes
Version	1.0
Workflow Status	Draft New – Specified as a parameter
Context	Specified as a parameter
Begin Date	Current Timestamp
Type	HAS_A
Source Low Cardinality	Derived from UML Association. The Source object is the class from which the link is drawn.
Source High Cardinality	Derived from UML Association. The Source object is the class from which the link is drawn.
Target Low Cardinality	Derived from UML Association. The Target object is the class to which the link is drawn.
Target High Cardinality	Derived from UML Association. The Target object is the class to which the link is drawn.
Direction	Navigability. Source-to-Target Target-to-Source Bidirectional

Table 4.12 New Object Class Relationship details

Mapping UML Inheritance

Each inheritance type association in the UML model is mapped to an Object Class Relationship in caDSR with the same attributes described for associations, except for Object Class Relationship Type, which for inheritance associations is “IS_A” instead of “HAS_A.”

Additionally, the child class of the association inherits all of the attributes of the parent class. DECs are created based on the combinations of the child class and each of its parent’s attributes. These combinations are mapped using the same process described in *Mapping Data Element Concepts* on page 107. Data elements are then created using the mapped DECs and the associated datatypes. These combinations are mapped using the same process described in *Mapping Data Elements* on page 108.

In addition, all DECs and DEs are appended with the appropriate classification and alternate name/definition information as described in *Relating (Classifying) Metadata Items to UML Models* on page 94.

Table 4.13 and *Table 4.14* below provide an overview of the information used to map the DECs and DEs for inheritance associations.

Data Element Concept Attribute	***Data***
Preferred Name	Child Object Class Public ID + Object Class Version: Parent Property Public ID + Property Version
Long Name	Child Object Class Long Name + Parent Property Long Name
Preferred Definition	Child Object Class Preferred Definition + Parent Property Preferred Definition
Version	1.0 (Specified as a parameter)
Workflow Status	Draft New (Specified as a parameter)
Context	Specified as a parameter
Begin Date	Current Timestamp
Object Class	Object Class corresponding to the Child UML Class
Property	Property corresponding to the Parent UML Attribute

Table 4.13 Inheritance Data Element Concept mapping

Data Element Attribute	***Data***
Preferred Name	DEC Public ID + DEC Version: Value Domain Public ID + Value Domain Version
Long Name	DEC Long Name + Value Domain Long Name
Preferred Definition	Derived from the underlying attribute description in the UML class diagram.
Version	1.0 – Specified as a parameter
Workflow Status	Draft New – Specified as a parameter
Context	Specified as a parameter
Begin Date	Current Timestamp
Data Element Concept	The Data Element Concept corresponding to the Child UML Class and the Parent Attribute.

Table 4.14 Inheritance Data Element mapping

Data Element Attribute	***Data***

Value Domain	Corresponds to the datatype of the Parent Attribute:
	java.lang.String = Value Domain Name “java.lang.String”
	java.lang.Boolean = Value Domain “java.lang.Boolean”
	java.lang.Long = Value Domain “java.lang.Boolean”
	java.lang.Integer = Value Domain “java.lang.Boolean”
	java.lang.Float = Value Domain “java.lang.Float”
	java.util.Date = Value Domain “java.lang.Date”
	int = Value Domain “int”
	long = Value Domain “long”
	boolean = Value Domain “boolean”
	char = Value Domain “char”
	double = Value Domain “double”
	float = Value Domain “float”
	byte = Value Domain “byte”
	short = Value Domain “short”

Table 4.14 Inheritance Data Element mapping

Reviewing and Verifying Registered Metadata

After your model is submitted for loading to caDSR, the model is first loaded to the caDSR “Sandbox.” The Sandbox functions as a dry-run environment, so that model owners and NCICB staff can be sure the model will load properly, and that the elements in the model are mapped to the appropriate caDSR metadata components. Once a successful load is verified, the UML model can be loaded to the caDSR Production database.

The NCICB provides two tools that allow you to search the caDSR and view loaded UML model elements: the UML Browser and the CDE Browser. Both of these tools are available for use on the Sandbox and Production environments. Model owners and NCICB content curators will work primarily with the Sandbox environment until the model is ready for Production. The steps for using the browser tools are the same, regardless of environment; the only differences are the URLs used to access the different repositories.

To access the browser tools for each environment/repository:

- Sandbox UML Model Browser: http://umlmodelbrowser-sandbox.nci.nih.gov
- Sandbox CDE Browser: https://cdebrowser-sandbox.nci.nih.gov
- Production UML Model Browser: http://umlmodelbrowser.nci.nih.gov
- Production CDE Browser: https://cdebrowser.nci.nih.gov

Since this section is designed to provide information regarding review and verification of newly loaded models, the procedures direct you to use the Sandbox URLs.

Note: The information in this section is provided solely as an overview to using the caDSR browser tools. For detailed information on using the UML Model Browser or the CDE Browser, see the respective online help for those tools.

Using the UML Model Browser

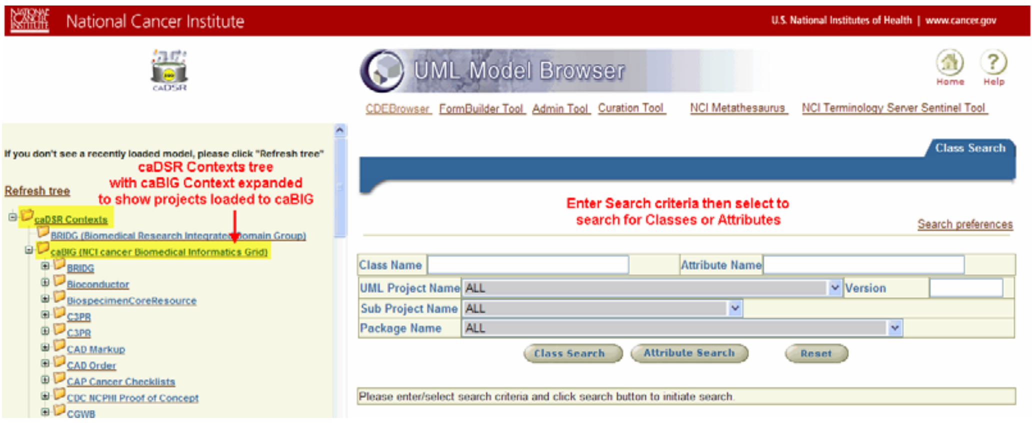
Since model owners are typically more familiar with the elements in their UML models than with the CDEs in caDSR, many find the UML Model Browser easier to use to find and verify the loading and mapping of the elements from their model.

The UML Model Browser allows you to browse the caDSR repository for all elements in your project, or search the repository using the class and class element names from your model. You can also select which project, sub-project and/or package to search specifically for your model’s elements.

The search result set provides links that open the CDE Browser, displaying the details of the caDSR administered components to which the elements are mapped.

To view registered UML model elements using the UML Model Browser:

1. Navigate to the following URL: http://umlmodelbrowser-sandbox.nci.nih.gov.


National Cancer Institute
U.S. National Institutes of Health | www.cancer.gov
UML Model Browser
Home Help
CDEBrowser FormBuilder Tool Admin Tool Curation Tool NCI Metathesaurus NCI Terminology Server Sentinel Tool

If you don't see a recently loaded model, please click "Refresh tree"
Refresh tree
caDSR Contexts tree with caBIG Context expanded to show projects loaded to caBIG

- caDSR Contexts
  - BRIDG (Biomedical Research Integrated Domain Group)
  - caBIG (NCI cancer Biomedical Informatics Grid)
    - BRIDG
    - Bioconductor
    - BiospecimenCoreResource
    - C3PR
    - C3PR
    - CAD Markup
    - CAD Order
    - CAP Cancer Checklists
    - CDC NCPHI Proof of Concept
    - CGWB

Class Search
Enter Search criteria then select to search for Classes or Attributes
Search preferences

Class Name		Attribute Name	
UML Project Name	ALL	Version	
Sub Project Name	ALL		
Package Name	ALL		

Class Search Attribute Search Reset

Please enter/select search criteria and click search button to initiate search.

Figure 4.6 UML Model Browser showing expanded caBIG Context and Search area

2. Within the UML Model Browser window, you have the following options for browsing elements loaded to the caDSR:
   - To view all of the loaded elements for a context, click the context name from the navigation tree on the left side of the window;
   - To view all of the loaded elements for your project, click the plus sign (+) to the left of the context to which your model was loaded (caBIG unless otherwise specified in your submission) and click your project's name from the expanded list.
   - Use the right side of the window to enter search criteria, if desired select the Project Name, Sub Project Name, or Package Name you want to search, and then click either **Class Search** or **Attribute Search**.

The sections below provide more information on how to browse, search for, and view model elements using the UML Model Browser.

Viewing all registered elements for a Context or Project

When the UML Model Browser first appears, the navigation tree on the left shows all of the contexts in caDSR in "collapsed" mode. Each of the items listed in the left navigation tree is an active link. Selecting an item in the tree instructs the UML Browser to return all of the loaded model elements associated with the selected item.

You must click the plus sign to the left of a context to expand it and show all of the projects that have been loaded to that context. You must click the plus sign to the left of a project name to expand it and show all of the sub projects and packages associated with that project.

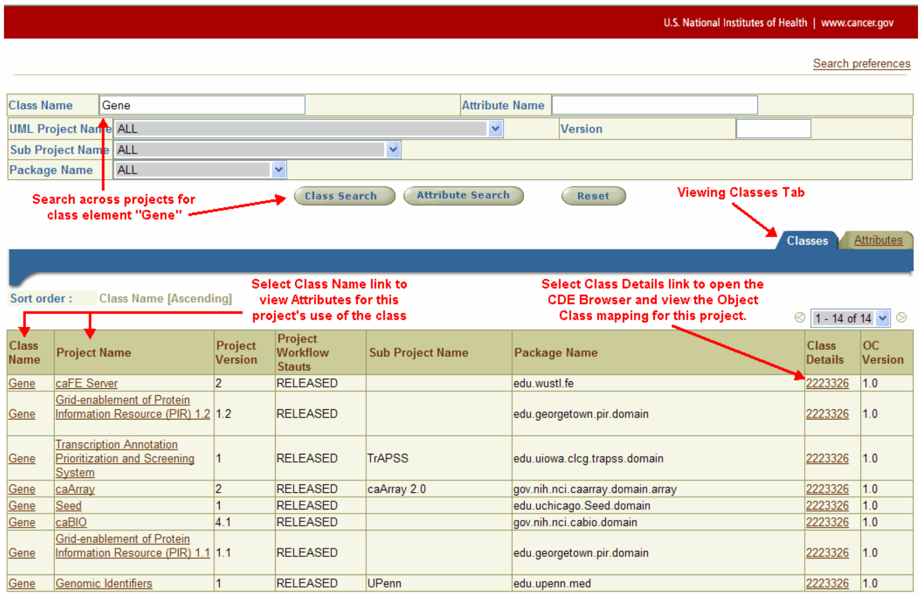
Expanding each node provides access to the active links for the items within the node, allowing you to view all of the elements for that node. This means you can view all loaded elements for a context, for a project, for a sub project or for a package, depending on the level of the node you select from the navigation tree.

Searching in the UML Model Browser

The UML Model Browser provides a straightforward search interface, allowing you to enter a class or attribute element name and then to select a project, sub project or package as well as version number to search for that class or attribute. However, it also allows you the flexibility to search for all attributes associated with a particular class, or for all classes containing a particular attribute.

To search for a class element:

1. Enter the element name in the **Class Name** text box. You may use an asterisk (*) as a wildcard if necessary.
2. If desired, use the drop-down lists to limit the search to a specific **UML Project**, **Sub Project** or **Package**, and identify a **Version** number if appropriate.
3. Click **Class Search**. The results appear below the search area.


U.S. National Institutes of Health | www.cancer.gov
Search preferences
Search across projects for class element "Gene"
Class Search
Attribute Search
Reset
Viewing Classes Tab
Classes
Attributes
Select Class Name link to view Attributes for this project's use of the class
Select Class Details link to open the CDE Browser and view the Object Class mapping for this project.
1 - 14 of 14

Class Name	Gene	Attribute Name	
UML Project Name	ALL	Version	
Sub Project Name	ALL		
Package Name	ALL		

Sort order : Class Name [Ascending]

Class Name	Project Name	Project Version	Project Workflow Stauts	Sub Project Name	Package Name	Class Details	OC Version
Gene	caFE Server	2	RELEASED		edu.wustl.fe	2223326	1.0
Gene	Grid-enablement of Protein Information Resource (PIR) 1.2	1.2	RELEASED		edu.georgetown.pir.domain	2223326	1.0
Gene	Transcription Annotation Prioritization and Screening System	1	RELEASED	TrAPSS	edu.uiowa.clcg.trapss.domain	2223326	1.0
Gene	caArray	2	RELEASED	caArray 2.0	gov.nih.nci.caarray.domain.array	2223326	1.0
Gene	Seed	1	RELEASED		edu.uchicago.Seed.domain	2223326	1.0
Gene	caBIO	4.1	RELEASED		gov.nih.nci.cabio.domain	2223326	1.0
Gene	Grid-enablement of Protein Information Resource (PIR) 1.1	1.1	RELEASED		edu.georgetown.pir.domain	2223326	1.0
Gene	Genomic Identifiers	1	RELEASED	UPenn	edu.upenn.med	2223326	1.0

Figure 4.7 UML Model Browser - Classes tab after search for class "Gene"

Conducting a **Class Search** against a **Class Name** returns all of the class elements and projects matching the criteria you entered. You can also enter an **Attribute Name** and a **Class Name** and click **Class Search**. In this instance, your results will be limited to only those occurrences of the class that contain the identified attribute.

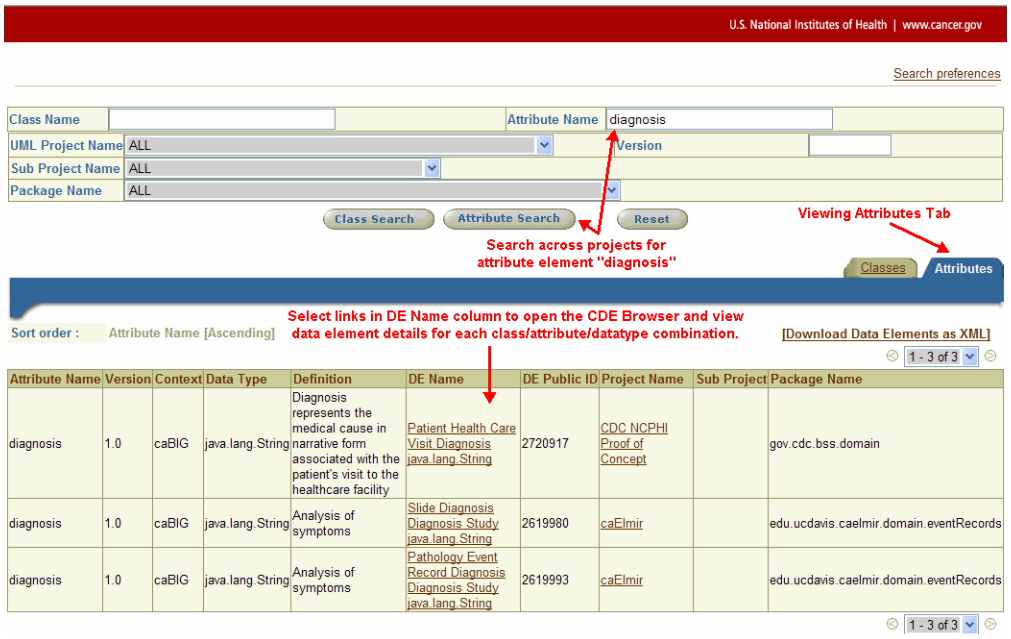
From the Classes tab you have several options for viewing additional information:

- Click a **Class Name** link - this displays all of the attributes of the class for the project listed.
- Click a **Project Name** link - this displays detailed information about the project.
- Click a **Class Details** link - this opens the CDE Browser and shows the Object Class information for the OC to which this class is mapped for this project.

You may also select to view the Attributes tab. The UML Browser then performs a new search for all of the attributes in caDSR that are associated with the class name entered (for the identified projects/sub projects/packages). This is the same result you would get if you entered a **Class Name** and clicked the **Attribute Search** button.

To search for an attribute element:

1. Enter the element name in the **Attribute Name** text box. You may use an asterisk (*) as a wildcard if necessary.
2. If desired, use the drop-down lists to limit the search to a specific **UML Project**, **Sub Project** or **Package**, and identify a **Version** number if appropriate.
3. Click **Attribute Search**. The results appear below the search area.


U.S. National Institutes of Health | www.cancer.gov
Search preferences
Class Name
Attribute Name
diagnosis
UML Project Name
ALL
Version
Sub Project Name
ALL
Package Name
ALL
Class Search
Attribute Search
Reset
Viewing Attributes Tab
Search across projects for attribute element "diagnosis"
Classes
Attributes
Select links in DE Name column to open the CDE Browser and view data element details for each class/attribute/datatype combination.
Sort order :
Attribute Name [Ascending]
[Download Data Elements as XML]
1 - 3 of 3

Attribute Name	Version	Context	Data Type	Definition	DE Name	DE Public ID	Project Name	Sub Project	Package Name
diagnosis	1.0	caBIG	java.lang.String	Diagnosis represents the medical cause in narrative form associated with the patient's visit to the healthcare facility	Patient Health Care Visit Diagnosis java.lang.String	2720917	CDC NCPHI Proof of Concept		gov.cdc.bss.domain
diagnosis	1.0	caBIG	java.lang.String	Analysis of symptoms	Slide Diagnosis Diagnosis Study java.lang.String	2619980	caElmir		edu.ucdavis.caelmir.domain.eventRecords
diagnosis	1.0	caBIG	java.lang.String	Analysis of symptoms	Pathology Event Record Diagnosis Diagnosis Study java.lang.String	2619993	caElmir		edu.ucdavis.caelmir.domain.eventRecords

Figure 4.8 UML Model Browser - Classes tab after search for class "Gene"

Conducting an **Attribute Search** against an **Attribute Name** returns all of the attribute elements and projects matching the criteria you entered. From the Attributes tab you can click a DE Name link for the element. This opens the CDE Browser and shows the data element information for the CDE to which this attribute (class/attribute/datatype combination) is mapped for the identified project.

You may also select to view the Classes tab. In this instance, the UML Browser performs a new search for all of the classes in caDSR that are associated with the attribute name entered (for the identified projects/sub projects/packages). This is the same result you would get if you entered an **Attribute Name** and clicked the **Class Search** button.

You may also choose to enter a **Class Name** with the **Attribute Name** and click **Attribute Search**. In this instance, your results will be limited to only those occurrences of the attribute that occur for the identified class.

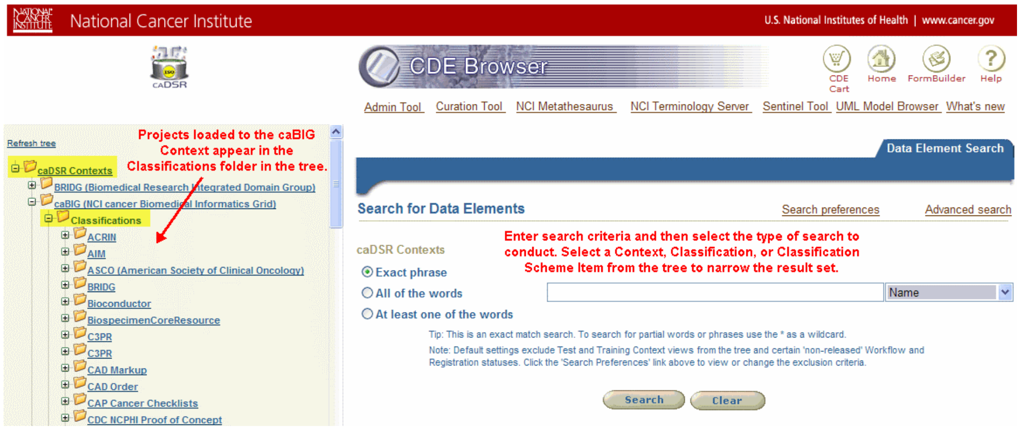
Using the CDE Browser

The CDE Browser allows you to browse for or search the caDSR repository for the administered components that have been registered through and mapped to the elements from your (or any other) project. The main difference between the CDE

Browser and the UML Model Browser is that the CDE Browser searches for and returns full data element information rather than specific class or attribute information for a project.

To view caDSR administered components using the CDE Browser:

1. Navigate to the following URL: https://cdebrowser-sandbox.nci.nih.gov.



Projects loaded to the caBIG Context appear in the Classifications folder in the tree.

- caDSR Contexts
  - BRIDG (Biomedical Research Integrated Domain Group)
  - caBIG (NCI cancer Biomedical Informatics Grid)
    - Classifications
      - ACRIN
      - AIM
      - ASCO (American Society of Clinical Oncology)
      - BRIDG
      - Bioconductor
      - BiospecimenCoreResource
      - C3PR
      - C3PR
      - CAD Markup
      - CAD Order
      - CAP Cancer Checklists
      - CDC NCPHI Proof of Concept

Search for Data Elements — Search preferences — Advanced search

Enter search criteria and then select the type of search to conduct. Select a Context, Classification, or Classification Scheme Item from the tree to narrow the result set.

caDSR Contexts

[✓ Exact phrase] All of the words At least one of the words

Name

Tip: This is an exact match search. To search for partial words or phrases use the * as a wildcard.

Note: Default settings exclude Test and Training Context views from the tree and certain 'non-released' Workflow and Registration statuses. Click the 'Search Preferences' link above to view or change the exclusion criteria.

Figure 4.9 CDE Browser showing expanded Classifications under the caBIG Context

2. Within the CDE Browser window, you have the following options for browsing the administered components within caDSR:
   - To view all of the CDEs for a context, click the context name from the navigation tree on the left side of the window;
   - To view all of the CDEs mapped to your project, click the plus sign (+) to the left of the context to which your model was loaded (caBIG unless otherwise specified in your submission) and then click the plus sign (+) to the left of the Classifications node. Then click your project's name from the expanded classifications list.
   - Use the right side of the window to enter search criteria, select the type of search to perform, and then click **Search**.

The sections below provide more information on how to browse, search for, and view data element details using the CDE Browser.

Viewing all CDEs for a Context, Classification Scheme or Classification Scheme Item

When the CDE Browser first appears, the navigation tree on the left shows all of the contexts in caDSR in "collapsed" mode. Each of the items listed in the left navigation tree is an active link. Selecting an item in the tree instructs the CDE Browser to return all of the CDEs associated with the selected item.

You must click the plus sign to the left of a context to expand it and show the sub-nodes, which include a Classifications node. All projects loaded to a context are "classified" and are listed as classifications under the context. You must click the plus sign to the left of the Classifications node to expand it and show all of the projects (classification schemes or CS) for the context. You must click the plus sign to the left of a CS (which corresponds to the project name) to expand it and show all of the sub projects and/or packages (classification scheme items or CSI) associated with that project.

Expanding each node provides access to the active links for the items within the node, allowing you to view all of the CDEs for that node. This means you can view all CDEs for a context, for a CS or for a CSI, depending on the level of the node you select from the navigation tree.

Searching in the CDE Browser

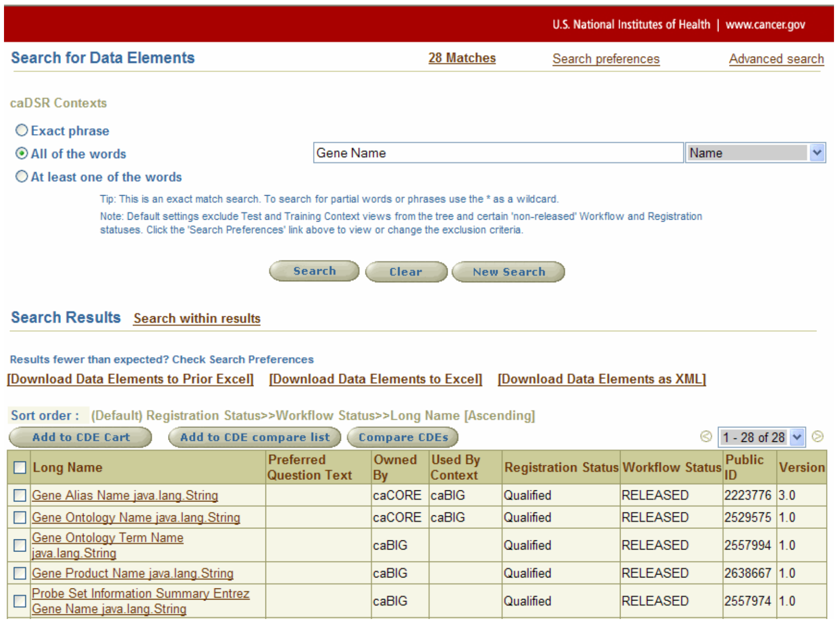
The CDE Browser provides a straightforward search interface, allowing you to enter search criteria text and then to select whether you want to perform an exact search, a search for all of the words or a search for any of the words.

Unless otherwise indicated, the CDE Browser will perform the search for entered criteria across all caDSR contexts.

To search for CDEs across all caDSR contexts:

1. In the search text box, enter the item you want to search for. You may use the asterisk (*) as a wildcard.
2. Select whether you want to search for an Exact Phrase, All of the words, or At least one of the words.
3. Click **Search**.

The search results appear in a list below the search area of the window, along with several feature buttons for manipulating the resulting CDEs and a smaller "paging" indicator that lists the number of results and which of those results are currently shown.



Exact phrase [✓ All of the words] At least one of the words

Gene Name | Name

Tip: This is an exact match search. To search for partial words or phrases use the * as a wildcard.

Note: Default settings exclude Test and Training Context views from the tree and certain 'non-released' Workflow and Registration statuses. Click the 'Search Preferences' link above to view or change the exclusion criteria.

Results fewer than expected? Check Search Preferences

[Download Data Elements to Prior Excel] [Download Data Elements to Excel] [Download Data Elements as XML]

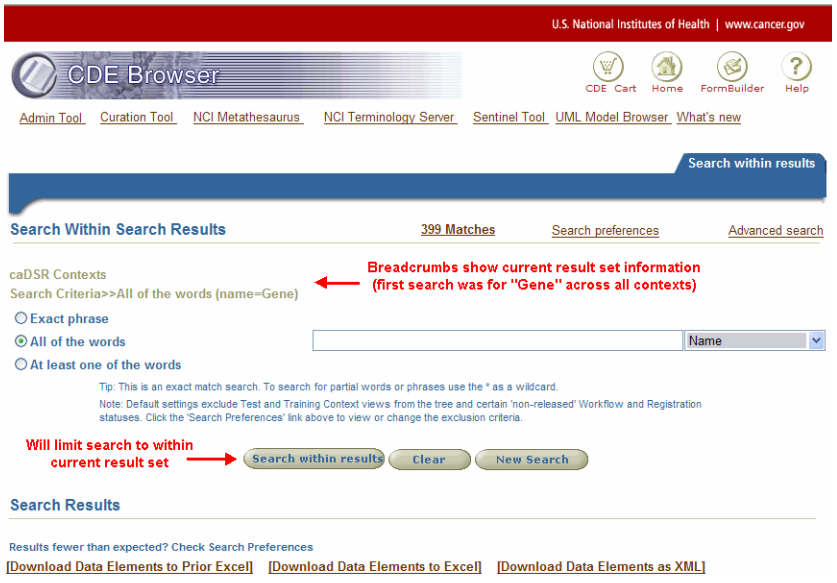
Sort order : (Default) Registration Status>>Workflow Status>>Long Name [Ascending]

☐	Long Name	Preferred Question Text	Owned By	Used By Context	Registration Status	Workflow Status	Public ID	Version
☐	Gene Alias Name java.lang.String		caCORE	caBIG	Qualified	RELEASED	2223776	3.0
☐	Gene Ontology Name java.lang.String		caCORE	caBIG	Qualified	RELEASED	2529575	1.0
☐	Gene Ontology Term Name java.lang.String		caBIG		Qualified	RELEASED	2557994	1.0
☐	Gene Product Name java.lang.String		caBIG		Qualified	RELEASED	2638667	1.0
☐	Probe Set Information Summary Entrez Gene Name java.lang.String		caBIG		Qualified	RELEASED	2557974	1.0

Figure 4.10 CDE Browser showing search criteria and result set

Clicking a Long Name link from the result set opens a detailed view of the data element, including additional information tabs and links to object class, classifications and other information.

If your result set is too large, you can click the Search Within Results link. This refreshes the search window, retaining the original result set, but changing the Search button to a Search Within Results button, and changing the breadcrumbs to indicate the subset of CDEs you will be searching.



caDSR Contexts
Search Criteria>>All of the words (name=Gene)

Exact phrase [✓ All of the words] At least one of the words

Tip: This is an exact match search. To search for partial words or phrases use the * as a wildcard.
Note: Default settings exclude Test and Training Context views from the tree and certain 'non-released' Workflow and Registration statuses. Click the 'Search Preferences' link above to view or change the exclusion criteria.

Results fewer than expected? Check Search Preferences
[Download Data Elements to Prior Excel] [Download Data Elements to Excel] [Download Data Elements as XML]

Figure 4.11 CDE Browser - searching within results

If you want to search for CDEs within a context, CS or CSI, you must first select that item from the navigation tree on the left. This populates the breadcrumbs with the name of the node and limits the CDE search to only those related items.

To search for CDEs within a context or classification:

1. From the navigation tree on the left of the CDE Browser window, click on the name of the grouping you want to search. You can click directly on a context to search that context, or you may have to expand (using the plus sign) the context and classification nodes to get to the name of the CS or CSI you want to search.
2. After the result set shows the CDEs for the node you selected and the breadcrumbs display the node you want to search, enter the search criteria into the text box, and select the proper search type (exact, all words, any words).
3. Click **Search**.

The new result set appears and the breadcrumbs change to reflect the new search used to create the currently displayed result set.

GLOSSARY

This glossary describes acronyms, objects, tools and other terms referred to in the chapters or appendixes of this guide.

Term	***Definition***
`{home_directory}`	Indicates the directory where you installed the SDK
`{package_structure}`	Indicates the package structure from the UML models
`{project_name}`	Indicates the `project_name` specified in the `deploy.properties` file
AGT	Artifact Generation Tool
AOP	Aspect Oriented Programming
API	Application Programming Interface
Writeable API	Methods exposed by the CSM to create, update and delete a domain object. These methods are generated using the code generation component.
Application Service	This refers to the CSM interface which exposes all the writeable as well as business methods for a particular application
BO	Business Object
C3D	Cancer Centralized Clinical Database
caBIG	cancer Biomedical Informatics Grid
caBIO	Cancer Bioinformatics Infrastructure Objects
caCORE	cancer Common Ontologic Representation Environment
caDSR	Cancer Data Standards Repository
caMOD	Cancer Models Database
cardinality	Cardinality describes the minimum and maximum number of associated objects within a set
CASE	Computer Aided Software Engineering
CCR	Center of Cancer Research
CDE	Common Data Element
CGAP	Cancer Genome Anatomy Project

Term	***Definition***
CMAP	Cancer Molecular Analysis Project
CODEGEN	Code generator tool
Code Generator Tool	The SDK tool that leverages Model-Driven Architecture to convert a UML model to a fully-functioning caCORE-like system
CS	Classification Scheme
CSI	Classification Scheme Item
CSM	Common Security Module
CTEP	Cancer Therapy Evaluation Program
CVS	Concurrent Versions System
DAO	Data Access Objects
DAS	Distributed Annotation System
DCP	Division of Cancer Prevention
DDL	Data Definition Language
DEC	Data Element Concept
DOM	Document Object Model
DTD	Document Type Definition
DU	Deployment Unit
EA	Enterprise Architect
EBI	European Bioinformatics Institute
EMF	Eclipse Modeling Framework
EVS	Enterprise Vocabulary Services
FK	Foreign Key - A collection of columns (attributes) that enforce a relationship to a Primary Key in another table used in data model tables in Enterprise Architect
FreeMarker	A "template engine"; a generic tool to generate text output (anything from HTML or RTF to auto generated source code) based on templates
GAI	CGAP Genetic Annotation Initiative
GEDP	Gene Expression Data Portal
IDE	Integrated Development Environment
ISO	International Organization for Standardization
JAF	JavaBeans Activation Framework
Jalopy	Source code formatting tool for the Sun Java Programming Language (http://jalopy.sourceforge.net/manual.html)
JAR	Java Archive
Javadoc	Tool for generating API documentation in HTML format from doc comments in source code (http://java.sun.com/j2se/javadoc/)
JDBC	Java Database Connectivity

Term	***Definition***
JDiff	Javadoc doc-let which generates an HTML report of all the packages, classes, constructors, methods, and fields which have been removed, added or changed in any way, including their documentation, when two APIs are compared (http://javadiff.sourceforge.net/)
JET	Java Emitter Templates
JMI	Java Metadata Interface
JSP	JavaServer Pages
JUnit	A simple framework to write repeatable tests (http://junit.sourceforge.net/)
MDR	Metadata Repository
metadata	Definitional data that provides information about or documentation of other data.
MMHCC	Mouse Models of Human Cancers Consortium
multiplicity	Multiplicity of an association end indicates the number of objects of the class on that end that may be associated with a single object of the class on the other end
MVC	Model-View-Controller, a design pattern
NCI	National Cancer Institute
NCICB	National Cancer Institute Center for Bioinformatics
NSC	Nomenclature Standards Committee
navigability	Navigability defines the visibility of an object to its associated source/target object at the other end of an association. Navigability is the same as directionality.
OMG	Object Management Group
OR	Object Relation
ORM	Object Relational Mapping
PCDATA	Parsed Character DATA
persistence layer	Data storage layer, usually in a relational database system
PK	Primary Key – Key used to uniquely identify a data model table in Enterprise Architect.
QA	Quality Assurance
RDBMS	Relational Database Management System
RUP	Rational Unified Process
SCM	Software Configuration Management
SDK	Software Development Kit
Semantic Connector	The SDK tool that links model elements to NCICB EVS concepts.
SPORE	Specialized Programs of Research
SQL	Structured Query Language

Term	***Definition***
Tagged value	A UML construct that represents a name-value pair; can be attached to anything in a UML model. Often used by UML modeling tools to store tool-specific information
UML	Unified Modeling Language
UML Loader	SDK tool that converts a domain model in UML to corresponding common data elements (CDEs) in caDSR.
UPT	User Provisioning Tool
URI	Uniform Resource Identifier
URL	Uniform Resource Locators
WSDL	Web Services Description Language
XMI	XML Metadata Interchange (http://www.omg.org/technology/documents/formal/xmi.htm) - The main purpose of XMI is to enable easy interchange of metadata between modeling tools (based on the OMG-UML) and metadata repositories (OMG-MOF) in distributed heterogeneous environments
XML	Extensible Markup Language (http://www.w3.org/TR/REC-xml/) - XML is a subset of Standard Generalized Markup Language (SGML). Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML
XP	Extreme Programming

Index