

[Home](#)[Knowledge Centers](#)[Discussion Forums](#)[Bugs/Feature Requests](#)[Development Code](#)[Repository](#)[page](#)[discussion](#)[view source](#)[history](#)

LexEVS Code Examples

[LexEVS 5.x Pick List Service](#) > [LexEVS 5.x caCORE Data Service API](#) > [LexEVS 5.x Analytical Grid Service API](#) > [LexEVS 5.x Data Grid Service API](#) > [LexEVS Code Examples](#)Contents [\[hide\]](#)

- 1 Introduction
- 2 LexEVS REST Examples
 - 2.1 Get CodingScheme details
 - 2.2 Get Concept by Code
 - 2.3 Get Concept by Code with wildcard
 - 2.4 Get Concept by Code by EntityDescription
 - 2.5 Get Concept by Code by EntityDescription with wildcard
 - 2.6 Get Concept by Presentation
 - 2.7 Get Concept by Presentation with wildcard
 - 2.8 Get Concept by Definition with wildcard
 - 2.9 Show Presentations of a Concept
 - 2.10 Show Definitions of a Concept
 - 2.11 Show Supported Associations for a CodingScheme
- 3 LexEVS Sample Code
 - 3.1 BuildTreeForCode
 - 3.2 CodingSchemeSelectionMenu
 - 3.3 FindCodesForDescription
 - 3.4 FindDescriptionForCode
 - 3.5 FindPropsAndAssocForCode
 - 3.6 FindRelatedCodes
 - 3.7 FindRelatedCodesWithPropertyLinks
 - 3.8 FindRelatedNodesForTermAndAssoc
 - 3.9 FindUMLSContextsForCUI
 - 3.10 ListHierarchy
 - 3.11 ListHierarchyByCode
 - 3.12 ListHierarchyMetaBySource
 - 3.13 ListHierarchyPathToRoot
 - 3.14 MetaDataSearch
 - 3.15 MetaMatch
 - 3.16 ProfileScheme
 - 3.17 ScoredIterator
 - 3.18 ScoredTerm
 - 3.19 ScoreTerm
 - 3.20 SoundsLike
 - 3.21 Util
- 4 LexEVS Sample Code in Distributed Environment
 - 4.1 Requirements
 - 4.2 Resources
 - 4.3 Example Code
 - 4.4 Example Test

vocabkc contents

- [Main Page](#)
- [What's New](#)
- [Forums](#)
- [Bugzilla](#)
- [Code Repository](#)
- [Feedback](#)
- [Contact Us](#)

tools

- [LexBIG/LexEVS](#)
- [LexWiki](#)
- [NCI Protégé](#)
- [Related Tools and Models](#)

projects

- [LexAjax](#)
- [LexGrid](#)
- [Cancer Data Standards Repository \(caDSR\)](#)
- [Common Terminology Criteria for Adverse Events \(CTCAE\)](#)
- [Open Health Natural Language Processing \(OHNLP\) Consortium](#)
- [Ontology Development and Information Extraction \(ODIE\)](#)

semantic infrastructure

- [SI Main Page](#)
- [Initiatives](#)
- [Requirements](#)

other resources

- [Library of Documents](#)
- [Documentation and Training for Tools](#)
- [Index of Terminologies](#)
- [Standards and Standards Influencing Organizations](#)
- [Outreach](#)

external links

- [VCDE Workspace](#)
- [caBIG@ Community](#)

Introduction

Website
 ■ caBIG® Support
 Service Providers

help

■ Editing Wiki Pages
 ■ Editing Forum Posts
 ■ Contact Us

search

toolbox

■ What links here
 ■ Related changes
 ■ Upload file
 ■ Special pages
 ■ Printable version
 ■ Permanent link
 ■ Print as PDF

This document is a section of the [Programmer's Guide](#).

Code examples provided here are to demonstrate LexEVS functionality.

LexEVS REST Examples

Get CodingScheme details

[http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.codingSchemes.CodingScheme&org.LexGrid.codingSchemes.CodingScheme&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

[query=org.LexGrid.codingSchemes.CodingScheme&org.LexGrid.codingSchemes.CodingScheme&codingSchemeName=Zebrafish&codingSchemeVersion=1.2](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.codingSchemes.CodingScheme&org.LexGrid.codingSchemes.CodingScheme&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

Get Concept by Code

[http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Concept&org.LexGrid.concepts.Concept[@_entityCode=ZFA_0001234]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

[query=org.LexGrid.concepts.Concept&org.LexGrid.concepts.Concept\[@_entityCode=ZFA_0001234\]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Concept&org.LexGrid.concepts.Concept[@_entityCode=ZFA_0001234]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

Get Concept by Code with wildcard

[http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Concept&org.LexGrid.concepts.Concept[@_entityCode=ZFA_000123*]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

[query=org.LexGrid.concepts.Concept&org.LexGrid.concepts.Concept\[@_entityCode=ZFA_000123*\]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Concept&org.LexGrid.concepts.Concept[@_entityCode=ZFA_000123*]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

Get Concept by Code by EntityDescription

[http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Concept,org.LexGrid.commonTypes.EntityDescription&org.LexGrid.commonTypes.EntityDescription[@_content=heart]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

[query=org.LexGrid.concepts.Concept,org.LexGrid.commonTypes.EntityDescription&org.LexGrid.commonTypes.EntityDescription\[@_content=heart\]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Concept,org.LexGrid.commonTypes.EntityDescription&org.LexGrid.commonTypes.EntityDescription[@_content=heart]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

Get Concept by Code by EntityDescription with wildcard

[http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Concept,org.LexGrid.commonTypes.EntityDescription&org.LexGrid.commonTypes.EntityDescription[@_content=he*r*]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

[query=org.LexGrid.concepts.Concept,org.LexGrid.commonTypes.EntityDescription&org.LexGrid.commonTypes.EntityDescription\[@_content=he*r*\]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Concept,org.LexGrid.commonTypes.EntityDescription&org.LexGrid.commonTypes.EntityDescription[@_content=he*r*]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

Get Concept by Presentation

[http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=Concept,Presentation,Text&Text\[@_content=heart\]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=Concept,Presentation,Text&Text[@_content=heart]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

Get Concept by Presentation with wildcard

[http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=Concept,Presentation,Text&Text\[@_content=heart*\]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=Concept,Presentation,Text&Text[@_content=heart*]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

Get Concept by Definition with wildcard

[http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=Concept,Definition,Text&Text\[@_content=Bilateral%20groups%20of%20cells*\]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=Concept,Definition,Text&Text[@_content=Bilateral%20groups%20of%20cells*]&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

Show Presentations of a Concept

[http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Presentation&org.LexGrid.concepts.Concept[@_entityCode=ZFA_0001234]&roleName=_presentationList&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

[query=org.LexGrid.concepts.Presentation&org.LexGrid.concepts.Concept\[@_entityCode=ZFA_0001234\]&roleName=_presentationList&codingSchemeName=Zebrafish&codingSchemeVersion=1.2](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Presentation&org.LexGrid.concepts.Concept[@_entityCode=ZFA_0001234]&roleName=_presentationList&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

Show Definitions of a Concept

[http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Definition&org.LexGrid.concepts.Concept[@_entityCode=ZFA_0001234]&roleName=_definitionList&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

[query=org.LexGrid.concepts.Definition&org.LexGrid.concepts.Concept\[@_entityCode=ZFA_0001234\]&roleName=_definitionList&codingSchemeName=Zebrafish&codingSchemeVersion=1.2](http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.concepts.Definition&org.LexGrid.concepts.Concept[@_entityCode=ZFA_0001234]&roleName=_definitionList&codingSchemeName=Zebrafish&codingSchemeVersion=1.2)

Show Supported Associations for a CodingScheme

<http://lexevsapi.nci.nih.gov/lexevsapi50/GetHTML?query=org.LexGrid.naming.SupportedAssociation&org.LexGrid.naming.SupportedAssociation&codingSchemeName=Zebrafish&codingSchemeVersion=1.2>

LexEVS Sample Code

The following examples are provided in the local installer in the 'examples' directory.

BuildTreeForCode

Attempts to provide a tree, based on a focus code, that includes the following information:

- All paths from the hierarchy root to one or more focus codes.
-

- Immediate children of every node in path to root
- Indicator to show whether any unexpanded node can be further expanded

This example accepts two parameters... The first parameter is required, and must contain at least one code in a comma-delimited list. A tree is produced for each code. Time to produce the tree for each code is printed in milliseconds. In order to factor out costs of startup and shutdown, resolving multiple codes may offer a better overall estimate performance.

The second parameter is optional, and can indicate a hierarchy ID to navigate when resolving child nodes. If not provided, "is_a" is assumed.

[CodingSchemeSelectionMenu](#)

Displays a list of available coding schemes.

[FindCodesForDescription](#)

Example showing how to find codes matching descriptive text. The program accepts up to two parameters...

The first param (required) indicates the text used to search matching descriptions. Matches are determined through a customized match algorithm, which uses a simple heuristic to try and rank returned values by relevance.

The second param (optional) indicates the type of entity to search. Possible values include the LexGrid built-in types "concept" and "instance". Additional supported types can be defined uniquely to a coding scheme. If provided, this should be a comma-delimited list of types. If not provided, all entity types are searched.

Example: FindCodesForDescription "blood" Example: FindCodesForDescription "breast cancer" "concept"

[FindDescriptionForCode](#)

Example showing how to find the entity description assigned to a specific code. The program accepts one parameter, the entity code.

[FindPropsAndAssocForCode](#)

Example showing how to find concept properties and associations based on a code.

[FindRelatedCodes](#)

Example showing how to find all concepts codes related to another code with distance 1.

[FindRelatedCodesWithPropertyLinks](#)

Example showing how to find all concepts codes related to another code with distance 1, plus the Property Link relations.

[FindRelatedNodesForTermAndAssoc](#)

Example showing how to find all endpoints of a named association for which the given term matches as source or target.

Note: the match algorithm applied to the term is the standard lucene query syntax.

[FindUMLSContextsForCUI](#)

Example showing any source-asserted hierarchies (based on import of MRHIER HCD) for a CUI. The program takes a single argument (the UMLS CUI), prompts for the code system to query in the LexGrid repository, and displays the available hierarchical relationships.

[ListHierarchy](#)

Example showing how to determine and display an unsorted list of root and subsumed nodes, up to a specified depth, for hierarchical relationships.

This program accepts two parameters:

The first parameter indicates the depth to display for the hierarchy. If 1, nodes immediately subsumed by the root are displayed. If 2, grandchildren are displayed, etc. If absent or < 0, a default depth of 3 is assumed.

The second parameter optionally indicates a specific hierarchy to navigate. If provided, this must match a registered identifier in the coding scheme supported hierarchy metadata. If left unspecified, all hierarchical associations are navigated. If an incorrect value is specified, a list of supported values will be output for future reference.

BACKGROUND: From a database perspective, LexBIG stores relationships internally in a forward direction, source to target. Due to differences in source formats, however, a wide variety of associations may be used ('PAR', 'CHD', 'isa', 'hasSubtype', etc). In addition, the direction of navigation may vary ('isa' expands in a reverse direction whereas 'hasSubtype' expands in a forward direction).

The intent of the `getHierarchy*` methods on the `LexBIGServiceConvenienceMethods` interface is to simplify the process of hierarchy discovery and navigation. These methods significantly reduce the need to understand conventions for root nodes, associations, and direction of navigation for a specific source format.

ListHierarchyByCode

Example showing how to determine and display the hierarchical relationships for a specific code, ancestors or descendants, within a fixed distance.

This program accepts two parameters, indicating the code and distance. The first parameter is the code (required). The second parameter is the distance (optional). If 1, immediate children are displayed. If 2, grandchildren are displayed, etc. If absent or < 0, all downstream branches are displayed.

BACKGROUND: From a database perspective, LexBIG stores relationships internally in a forward direction, source to target. Due to differences in source formats, however, a wide variety of associations may be used ('PAR', 'CHD', 'isa', 'hasSubtype', etc). In addition, the direction of navigation may vary ('isa' expands in a reverse direction whereas 'hasSubtype' expands in a forward direction).

The intent of the `getHierarchy*` methods on the `LexBIGServiceConvenienceMethods` interface is to simplify the process of hierarchy discovery and navigation. These methods significantly reduce the need to understand conventions for root nodes, associations, and direction of navigation for a specific source format.

ListHierarchyMetaBySource

Example showing how to determine and display an unsorted list of root and subsumed nodes, up to a specified depth, for hierarchical relationships. It is written specifically to handle display of relationships for a designated source within the NCI Metathesaurus.

This program accepts two parameters. The first indicates the depth to display hierarchical relations. If 0, only the root nodes are displayed. If 1, nodes immediately subsumed by the root are also displayed, etc. If < 0, a default depth of 0 is assumed.

The second parameter must provide the source abbreviation (SAB) of the Metathesaurus source to be evaluated (e.g. ICD9CM, MDR, SNOMEDCT).

ListHierarchyPathToRoot

Example showing how to determine and display paths from a given concept back to defined root nodes through any hierarchies registered for the coding scheme.

This program accepts one parameter (required), indicating the code to evaluate.

BACKGROUND: From a database perspective, LexBIG stores relationships internally in a forward direction, source to target. Due to differences in source formats, however, a wide variety of associations may be used ('PAR', 'CHD', 'isa', 'hasSubtype', etc). In addition, the direction of navigation may vary ('isa' expands in a reverse direction whereas 'hasSubtype' expands in a forward direction).

The intent of the `getHierarchy*` methods on the `LexBIGServiceConvenienceMethods` interface is to simplify the process of hierarchy discovery and navigation. These methods significantly reduce the need to understand conventions for root nodes, associations, and direction of navigation for a specific source format.

MetaDataSearch

Example how to query stored metadata for a code system. For the example, use the `LoadSampleMeta>DataData.bat` to load the required code system and metadata.

MetaMatch

Example attempting to approximate some characteristics of the Metaphrase search algorithm. However, full Metaphrase compatibility is not anticipated.

ProfileScheme

Requires loading valid scheme (must have root node named @ pointing to top nodes) Profiles a coding scheme based on unique URN, version, relation and scheme name.

Note: If the URN and version values are unspecified, a list of available coding schemes will be presented for user selection.

ScoredIterator

Used to wrap scored results for consumption as a standard `ResolvedConceptReferenceIterator`.

ScoredTerm

Used to manage and sort search results based on a scoring algorithm.

ScoreTerm

Example showing a simple scoring algorithm that evaluates a provided term against available terms in a code system. A cutoff percentage can optionally be provided.

SoundsLike

Example showing how to list concepts with presentation text that 'sounds like' a specified value.

Util

Utility functions to support the examples.

LexEVS Sample Code in Distributed Environment

This section of the document will discuss how to enable LexEVS example code for use in a LexEVS distributed environment.

The Distributed LexBIG (DLB) API is strictly a Java interface and requires Internet access for remote connectivity to the caCORE LexEVS server.

The example code in *LexEVS_51_Example_Code.zip* (provided for download at [LexEVS 5.1 Components](#)) can be modified to run in distributed mode with minimal changes.

Requirements

The following tasks are required in order to run in distributed mode.

Update classpath:

Access to the DLB API requires access to the *LexEVS_51_client.jar* file (available for download at [LexEVS 5.1 Components](#)), which needs to be available in the classpath.

Import the following into your java code:

```
import org.LexGrid.LexBIG.caCore.interfaces.LexEVSApplicationService;
import org.LexGrid.LexBIG.caCore.interfaces.LexEVSDataService;
import org.LexGrid.LexBIG.caCore.interfaces.LexEVSDataDistributed;
import org.LexGrid.LexBIG.caCore.interfaces.LexEVSService;
```


Update code to get LexBIG service:

```
//Set the LexEVS URL (for remote access)
String evsUrl = "http://lexevsapi.nci.nih.gov/lexevsapi50";

LexBIGService lbSvc;

// Get the LexBIG service reference from LexEVS Application Service
lbSvc = (LexEVSApplicationService)ApplicationServiceProvider.getApplicationServiceFromUrl(evsUrl, "EvsServiceInfo");
```

Resources

[Distributed Sample Client Code](#)  - ZIP archive file containing sample distributed code. (Refer to sample client code in /src folder)

Example Code

The following code example shows use of the DLB API to retrieve the list of available coding schemes in the LexEVS repository.

```

import java.util.List;

import gov.nih.nci.system.applicationservice.ApplicationException;
import gov.nih.nci.system.client.ApplicationServiceProvider;

import org.LexGrid.LexBIG.DataModel.Collections.CodingSchemeRenderingList;
import org.LexGrid.LexBIG.DataModel.InterfaceElements.CodingSchemeRendering;
import org.LexGrid.LexBIG.Exceptions.LBInvocationException;
import org.LexGrid.LexBIG.caCore.interfaces.LexEVSApplicationService;
import org.LexGrid.LexBIG.caCore.interfaces.LexEVSDDataService;
import org.LexGrid.LexBIG.caCore.interfaces.LexEVSDistributed;
import org.LexGrid.LexBIG.caCore.interfaces.LexEVSService;
import org.LexGrid.codingSchemes.CodingScheme;

public class RemoteTestClient {

    private String serviceUrl = "http://lexevsapi.nci.nih.gov/lexevsapi50";
    private LexEVSService lexevsService;

    public static void main(String args[]) throws Exception {
        RemoteTestClient client = new RemoteTestClient();

        client.queryLexEVSDistributed();
        //client.queryLexEVSDDataService();
    }

    public RemoteTestClient() throws Exception {
        lexevsService = (LexEVSApplicationService)ApplicationServiceProvider
            .getApplicationServiceFromUrl(serviceUrl, "EvsServiceInfo");
    }

    public void queryLexEVSDistributed() throws LBInvocationException {
        LexEVSDistributed distributedSvc = lexevsService;
        CodingSchemeRenderingList csrl = distributedSvc.getSupportedCodingSchemes();
        CodingSchemeRendering[] csr = csrl.getCodingSchemeRendering();

        for(int i=0;i<csr.length;i++){

            System.out.println("\t\tOutput: " + "Coding Scheme: " + csr[i].getCodingSchemeSummary().getLocalName());
            System.out.println("\t\tOutput: " + " -- Version: " + csr[i].getCodingSchemeSummary().getRepresentsVersion());
            System.out.println("\t\tOutput: " + " -- URI: " + csr[i].getCodingSchemeSummary().getCodingSchemeURI());
        }
    }

    public void queryLexEVSDDataService() throws ApplicationException {
        LexEVSDDataService dataSvc = lexevsService;
        CodingScheme codingScheme = new CodingScheme();
        List<CodingScheme> results = dataSvc.search(CodingScheme.class, codingScheme);

        for(CodingScheme cs : results){
            System.out.println("\t\tOutput: " + "Coding Scheme: " + cs.getLocalName());
            System.out.println("\t\tOutput: " + " -- Version: " + cs.getRepresentsVersion());
            System.out.println("\t\tOutput: " + " -- URI: " + cs.getCodingSchemeURI());
        }
    }
}

```

Example Test

The following test shows use of the DLB API to retrieve the list of available coding schemes in the LexEVS repository.

```

public class Test {
    /**
     * Initialize program variables
     */

    private String codingScheme = null;
    private String version = null;

    LexBIGService lbSvc;

    public Test(String codingScheme, String version) {
        //Set the LexEVS URL (for remote access)
        String evsUrl = "http://lexevsapi.nci.nih.gov/lexevsapi50/http/remoteService";
        boolean isRemote = true;
        this.codingScheme = codingScheme;
        this.version = version;

        // Get the LexBIG service reference from LexEVS Application Service
        lbSvc = (LexEVSApplicationService) ApplicationServiceProvider.getApplicationServiceFromUrl(evsUrl, "EvsServiceInfo");

        // Set the vocabulary to work with
        Boolean retval = adapter.setVocabulary(codingScheme);

        codingSchemeMap = new HashMap();
        try {
            // Using the LexBIG service, get the supported coding schemes
            CodingSchemeRenderingList csrl = lbSvc.getSupportedCodingSchemes();

            // Get the coding scheme rendering
            CodingSchemeRendering[] csrs = csrl.getCodingSchemeRendering();

```

```

// For each coding scheme rendering...
for (int i=0; i<csrs.length; i++) {
    CodingSchemeRendering csr = csrs[i];

    // Determine whether the coding scheme rendering is active or not
    Boolean isActive = csr.getRenderingDetail().getVersionStatus().equals(CodingSchemeVersionStatus.ACTIVE);
    if (isActive != null && isActive.equals(Boolean.TRUE)) {
        // Get the coding scheme summary
        CodingSchemeSummary css = csr.getCodingSchemeSummary();

        // Get the coding scheme formal name
        String formalname = css.getFormalName();

        //Get the coding scheme version
        String representsVersion = css.getRepresentsVersion();
        CodingSchemeVersionOrTag vt = new CodingSchemeVersionOrTag();
        vt.setVersion(representsVersion);

        // Resolve coding scheme based on the formal name
        CodingScheme scheme = null;

        try {
            scheme =lbSvc.resolveCodingScheme(formalname, vt);
            if (scheme != null) {
                codingSchemeMap.put((Object) formalname, (Object) scheme);
            }
        } catch (Exception e) {
            // Resolve coding scheme based on the URI
            String uri = css.getCodingSchemeURI();
            try {
                scheme = lbSvc.resolveCodingScheme(uri, vt);
                if (scheme != null) {
                    codingSchemeMap.put((Object) formalname, (Object) scheme);
                }
            } catch (Exception ex) {
                String localname = css.getLocalName();

                // Resolve coding scheme based on the local name
                try {
                    scheme = lbSvc.resolveCodingScheme(localname, vt);
                    if (scheme != null){
                        codingSchemeMap.put((Object) formalname, (Object) scheme);
                    }
                } catch (Exception e2) {
                }
            }
        }
    }
}

} catch (Exception e) {
    e.printStackTrace();
}

/**
 *Main
 */
public static void main (String[] args)
{
    String name = "NCI Thesaurus";
    String version = "06.12d";

    // Instantiate the Test Class
    Test test = new Test(name, version);
}

```

[Categories: VKC Contents](#) | [Documentation](#) | [LexEVS Code](#) | [LexEVS](#)



This page was last modified on 2 February 2010, at 04:58. This page has been accessed 1,145 times.

[CONTACT US](#) | [PRIVACY NOTICE](#) | [DISCLAIMER](#) | [ACCESSIBILITY](#) | [APPLICATION SUPPORT](#)

