

## LexEVS 5.x Value Domain Service

[LexEVS 5.x Design and Architecture Guide](#) [LexEVS Architecture](#) > [LexEVS 5.x Programmer's Guide](#) > [LexEVS 5.x API](#) > [LexEVS 5.x Query Service Extension](#) > [LexEVS 5.x Value Domain Service](#)

### 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 Website](#)
- [caBIG@ Support](#)

### Contents [\[hide\]](#)

- 1 [Introduction](#)
- 2 [Value Domain Class Diagram](#)
- 3 [LexEVS Value Domain Service API](#)
  - 3.1 [Administration Functions](#)
    - 3.1.1 [Loading Value Domain Definitions](#)
    - 3.1.2 [Remove Value Domain Definition](#)
    - 3.1.3 [Drop Value Domain tables](#)
  - 3.2 [Validate XML resources](#)
  - 3.3 [Query Functions](#)
    - 3.3.1 [isConceptInDomain](#)
    - 3.3.2 [resolveValueDomain](#)
    - 3.3.3 [isSubDomain](#)
    - 3.3.4 [getValueDomainDefinition](#)
    - 3.3.5 [listValueDomains](#)
    - 3.3.6 [getAllValueDomainsWithNoNames](#)
    - 3.3.7 [getValueDomainEntitiesForTerm](#)
    - 3.3.8 [getCodingSchemesInValueDomain](#)
    - 3.3.9 [isDomain](#)
  - 3.4 [Resolved Value Domain Objects](#)
    - 3.4.1 [ResolvedValueDomainCodedNodeSet](#)
    - 3.4.2 [ResolvedValueDomainDefinition](#)
  - 3.5 [Error Handling](#)
- 4 [Load Scripts](#)
- 5 [Sample Value Domain Definition XML file](#)
- 6 [System Testing](#)
- 7 [Installation / Packaging](#)

## Introduction

This document is a section of the [Programmer's Guide](#). It is new for LexEVS v5.1.

The Value Domain services are integrated parts of the LexEVS core API. They provide the ability to:

- load Value Domaindefinitions programmatically into the LexGrid repository using the domain objects that are available via the LexGrid logical model
- apply user restrictions (ex: valueDomain URI) and dynamically resolve the definitions at run time

The LexEVS Value Domain Services expose the API particularly for the Value Domain elements of the LexGrid Logical Model. For more information on LexGrid model see [\[1\]](#) [↗](#)

## Value Domain Class Diagram

These classes implement the LexEVS Value Domain API.

Class Name	Description
VDServices	Class to handle list of Value Domain Definitions Object to and fro database
VDServices	Class to handle individual Value Domain Definition objects to and fro database.
VDEntryServices	Class to handle Value Domain Entry objects to and fro database.
LexEVSValueDomainServices	Primary interface for LexEVS Value Domain API
LexEVSValueDomainServicesImpl	Implementation of LexEVSValueDomainServices which is primary interface for LexEVS Value Domain API.

## Service Providers

## help

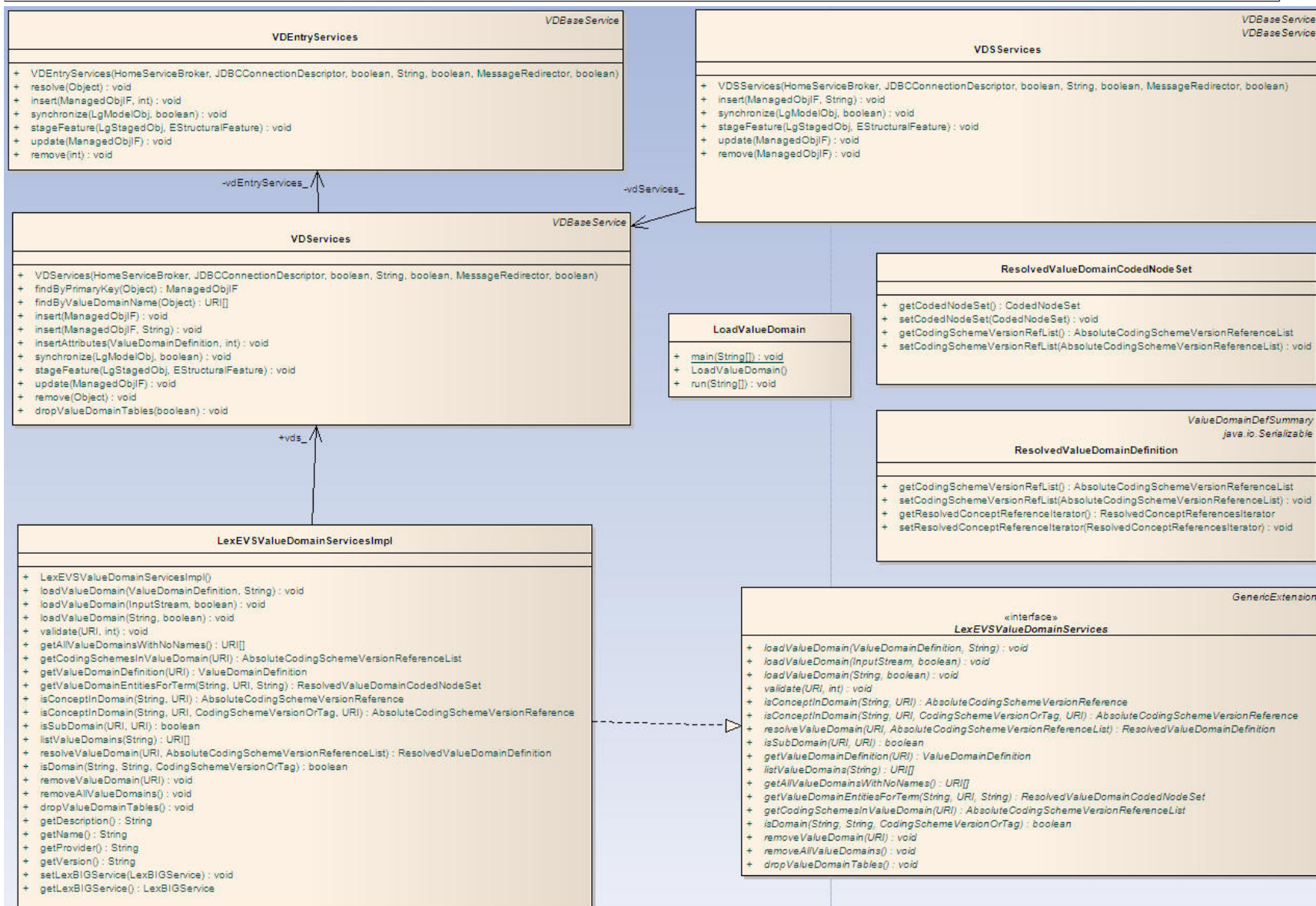
- [Editing Wiki Pages](#)
- [Editing Forum Posts](#)
- [Contact Us](#)

## search

## toolbox

- [What links here](#)
- [Related changes](#)
- [Upload file](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)
- [Print as PDF](#)

<b>LoadValueDomain</b>	Imports the value Domain Definitions in the source file, provided in LexGrid canonical format, to the LexBIG repository.
<b>ResolvedValueDomainCodedNodeSet</b>	Contains coding scheme version reference list that was used to resolve the value domain and the coded node set.
<b>ResolvedValueDomainDefinition</b>	A resolved Value Domain definition containing the coding scheme version reference list that was used to resolve the value domain and an iterator for resolved concepts.



## LexEVS Value Domain Service API

## Administration Functions

LexEVS Value Domain Services provide administration functions to load and remove value domain definitions.

## Loading Value Domain Definitions

There are three methods that could be used to load Value Domain Definitions:

- `loadValueDomain(ValueDomainDefinition vddef, String systemReleaseURI)`

- `loadValueDomain(InputStream inputStream,boolean failOnAllErrors)`
- `loadValueDomain(String xmlFileLocation, boolean failOnAllErrors)`

`loadValueDomain(ValueDomainDefinition vddef, String systemReleaseURI)`

<code>loadValueDomain(ValueDomainDefinition vddef, String systemReleaseURI)</code>	
Description:	Loads supplied valueDomainDefinition object.
Input:	<code>org.LexGrid.emf.valueDomains.ValueDomainDefinition</code> string
Output:	none
Exception:	<code>LBException</code>
Implementation Details:	<p><b>Implementation:</b></p> <p><i>Step 1:</i> Call this method on the associated LexEVS Value Domain Service instance to load a Value Domain Definition object and the System Release URI this definition belongs to.</p> <p><b>Sample Call:</b></p> <p><i>Step 1:</i> Instantiate <code>LexEVSVValueDomainServices</code> if it is not done yet :  <code>org.lexgrid.valuedomain.LexEVSVValueDomainServices vds = new org.lexgrid.valuedomain.impl.LexEVSVValueDomainServicesImpl();</code></p> <p><i>Step 2:</i> Create and populate the ValueDomainDefinition object. ValueDomainDefinition can be created using: <code>org.LexGrid.emf.valueDomains.ValueDomainDefinition valueDomain = org.LexGrid.emf.valueDomains.ValueDomainsFactory.eINSTANCE.createValueDomainDefinition();</code></p> <p>Data for valueDomain object can be populated by using set methods:  <code>valueDomain.setValueDomainURI(uri); valueDomain.setValueDomainName(name); valueDomain.setDefaultCodingScheme(cs); valueDomain.setEntityDescription(ed);</code></p> <p>Similarly, DefinitionEntry, Property, and Mapping objects can be created and assigned to the valueDomain object: <code>valueDomain.getDefinitionEntry.add(vdEntry); valueDomain.setProperties(propertiesObject); valueDomain.setMappings(mappingsObject);</code></p> <p><i>Step 3:</i> Call the load method by passing the Value Domain Definition object and the System Release URI: <code>vds.loadvalueDomain(valueDomain,"Release 2009");</code></p>

`loadValueDomain(InputStream inputStream,boolean failOnAllErrors)`

<code>loadValueDomain(InputStream inputStream,boolean failOnAllErrors)</code>	
Description:	Loads valueDomainDefinitions found in inputStream.
Input:	<code>java.io.InputStream</code> boolean
Output:	none
Exception:	<code>Exception</code>
Implementation Details:	<p><b>Implementation:</b></p> <p><i>Step 1:</i> Call this method on the associated LexEVS Value Domain Service instance to load all Value Domain Definitions from the inputStream.</p> <p><b>Sample Call:</b></p> <p><i>Step 1:</i> Instantiate <code>LexEVSVValueDomainServices</code> if it is not done yet :  <code>org.lexgrid.valuedomain.LexEVSVValueDomainServices vds = new org.lexgrid.valuedomain.impl.LexEVSVValueDomainServicesImpl();</code></p> <p><i>Step 2:</i> Call load method by passing the inputStream and boolean flag whether to stop on load errors:  <code>vds.loadvalueDomain(inputStream, true);</code></p>

`loadValueDomain(String xmlFileLocation, boolean failOnAllErrors)`

<code>loadValueDomain(String xmlFileLocation, boolean failOnAllErrors)</code>	
---	--

Description:	Loads valueDomainDefinitions found in input xml file.
Input:	<i>java.lang.String</i> boolean
Output:	none
Exception:	<i>Exception</i>
Implementation Details:	<p><b>Implementation:</b></p> <p><i>Step 1:</i> Call this method on the associated LexEVS Value Domain Service instance to load all Value Domain Definitions found in an XML file that is in LexGrid format.</p> <p><b>Sample Call:</b></p> <p><i>Step 1:</i> Instantiate LexEVSValueDomainServices if it is not done yet :  <code>org.lexgrid.valuedomain.LexEVSValueDomainServices vds = new  org.lexgrid.valuedomain.impl.LexEVSValueDomainServicesImpl();</code></p> <p><i>Step 2:</i> Call load method by passing the inputfile location and boolean flag whether to stop on load errors: <code>vds.loadvalueDomain(inputXMLFile, true);</code></p>

#### Remove Value Domain Definition

Below are the functions to remove value domain definitions from the system.

#### removeValueDomain(URI valueDomainURI)

removeValueDomain(URI valueDomainURI)	
Description:	Removes supplied value domain definition from the system.
Input:	<i>java.net.URI</i>
Output:	none
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException,</i> <i>org.LexGrid.managedobj.RemoveException</i>
Implementation Details:	<p><b>Implementation:</b></p> <p><i>Step 1:</i> Call this method on the associated LexEVS Value Domain Service instance to remove the Value Domain Definition from the system that matches the supplied URI.</p> <p><b>Sample Call:</b></p> <p><i>Step 1:</i> Instantiate LexEVSValueDomainServices if it is not done yet :  <code>org.lexgrid.valuedomain.LexEVSValueDomainServices vds = new  org.lexgrid.valuedomain.impl.LexEVSValueDomainServicesImpl();</code></p> <p><i>Step 2:</i> Call removeValueDomain method <code>vds.removeValueDomain (new  URI("AUTO:AllDomesticANDGM"));</code></p>

#### removeAllValueDomains()

removeAllValueDomains()	
Description:	Removes all value domain definitions from the system.
Input:	none
Output:	none
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException,</i> <i>org.LexGrid.managedobj.RemoveException</i>
Implementation Details:	<p><b>Implementation:</b></p> <p><i>Step 1:</i> Call this method on the associated LexEVS Value Domain Service instance to remove all the Value Domain Definitions that are loaded in the system.</p> <p><b>Sample Call:</b></p>

*Step 1:* Instantiate LexEVSVValueDomainServices if it is not done yet :  
`org.lexgrid.valuedomain.LexEVSVValueDomainServices vds = new  
org.lexgrid.valuedomain.impl.LexEVSVValueDomainServicesImpl();`  
*Step 2:* Call removeAllValueDomains method `vds.removeAllValueDomains();`

#### Drop Value Domain tables

Administration function to delete value domain and pick list tables. Tables will be deleted only if there are no contents in value domain and pick list tables.

`dropValueDomainTables()`

<code>dropValueDomainTables()</code>	
Description:	Drops value domain tables only if there are no value domain and pick list entries.
Input:	none
Output:	none
Exception:	<code>org.LexGrid.LexBIG.Exceptions.LBException</code> , <code>org.LexGrid.managedobj.RemoveException</code>
Implementation Details:	<p><b>Implementation:</b></p> <p><i>Step 1:</i> Call this method on the associated LexEVS Value Domain Service instance to drop both Value Domain and Pick List tables. The tables will be dropped only if there are no entries in both Value Domain and Pick List tables.</p> <p><b>Sample Call:</b></p> <p><i>Step 1:</i> Instantiate LexEVSVValueDomainServices if it is not done yet :  <code>org.lexgrid.valuedomain.LexEVSVValueDomainServices vds = new  org.lexgrid.valuedomain.impl.LexEVSVValueDomainServicesImpl();</code>  <i>Step 2:</i> Call dropValueDomainTables method <code>vds.dropValueDomainTables();</code></p>

#### Validate XML resources

Function to perform validation of source XML file without loading.

`validate(URI uri, int valicationLevel)`

<code>validate(URI uri, int valicationLevel)</code>	
Description:	Performs validation of the candidate resource without loading data.
Input:	<code>java.net.URI</code> int
Output:	none
Exception:	<code>Org.LexGrid.LexBIG.Exceptions.LBParameterException</code>
Implementation Details:	<p><b>Implementation:</b></p> <p><i>Step 1:</i> Call this method on the associated LexEVS Value Domain Service instance to validate the XML file that is in LexGrid format. This call will not load the data in XML file.</p> <p><b>Sample Call:</b></p> <p><i>Step 1:</i> Instantiate LexEVSVValueDomainServices if it is not done yet :  <code>org.lexgrid.valuedomain.LexEVSVValueDomainServices vds = new  org.lexgrid.valuedomain.impl.LexEVSVValueDomainServicesImpl();</code>  <i>Step 2:</i> Call validate method for validation by supplying URI of the XML file and validation level.  Supported validationLevels includes:  0 = Verify document is well-formed  1 = Verify document is valid  <code>vds.validaten(uriOfXMLFile, true);</code></p>

#### Query Functions

Here are some of the query functions that can be run against value domain definitions using the LexEVS Value Domain Services.

## isConceptInDomain

There are two methods to check if supplied concept code is part of supplied value domain definition.

[isConceptInDomain\(String entityCode, URI valueDomainURI\)](#)

isConceptInDomain(String entityCode, URI valueDomainURI)	
Description:	Determines if the supplied entity code is a valid result for the supplied value domain and, if it is, returns the particular codingSchemeVersion that was used.
Input:	<i>java.lang.String</i> , <i>java.net.URI</i>
Output:	<i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p><b>Implementation:</b></p> <p><i>Step 1:</i> Call this method on the associated LexEVS Value Domain Service instance to determine if the supplied entity code is a valid in the supplied Value Domain URI. If it is, returnS Coding Scheme URI and the Version that was used to resolve; otherwise, returns null.</p> <p><b>Sample Call:</b></p> <p><i>Step 1:</i> Instantiate LexEVSValueDomainServices if it is not done yet :  <i>org.lexgrid.valuedomain.LexEVSValueDomainServices vds = new</i>  <i>org.lexgrid.valuedomain.impl.LexEVSValueDomainServicesImpl();</i></p> <p><i>Step 2:</i> Call isConceptInComdin method: <i>AbsoluteCodingSchemeVersionReference acsvr = vds.isConceptInDomain("conceptA", "valueDomainURI");</i></p>

[isConceptInDomain\(String entityCode, URI entityCodeNamespace, CodingSchemeVersionOrTag csvt, URI valueDomainURI\)](#)

isConceptInDomain(String entityCode, URI entityCodeNamespace, CodingSchemeVersionOrTag csvt, URI valueDomainURI)	
Description:	Similar to the previous method, this method determines if the supplied entity code and entity namespace is a valid result for the supplied value domain when resolved against the supplied Coding Scheme Version or Tag.
Input:	<i>java.lang.String</i> , <i>java.net.URI</i> , <i>org.LexGrid.LexBIG.DataModel.Core.CodingSchemeVersionOrTag</i> <i>java.net.URI</i>
Output:	<i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p><b>Implementation:</b></p> <p><i>Step 1:</i> Call this method on the associated LexEVS Value Domain Service instance to determine if the supplied entity code and its entity namespace is valid in the supplied Value Domain URI when resolved against supplied Coding Scheme Version or Tag. If it is, return Coding Scheme URI and the Version that was used to resolve; otherwise, returns null.</p> <p><b>Sample Call:</b></p> <p><i>Step 1:</i> Instantiate LexEVSValueDomainServices if it is not done yet :  <i>org.lexgrid.valuedomain.LexEVSValueDomainServices vds = new</i>  <i>org.lexgrid.valuedomain.impl.LexEVSValueDomainServicesImpl();</i></p> <p><i>Step 2:</i> Call isConceptInDomain method: <i>AbsoluteCodingSchemeVersionReference acsvr = vds.isConceptInDomain("conceptA", "conceptAEntityNamespace", "codingSchemeVersion", "valueDomainURI");</i></p>

## resolveValueDomain

This method resolves supplied value domain definition and returns set of valied concept codes plus the code systems and its versions that were used.

[resolveValueDomain\(URI valueDomainURI, AbsoluteCodingSchemeVersionReferenceList csVersionList\)](#)

[resolveValueDomain\(URI valueDomainURI, AbsoluteCodingSchemeVersionReferenceList](#)

csVersionList)	
Description:	Resolves a value domain using the supplied set of coding scheme versions.
Input:	<i>java.net.URI</i> , <i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReferenceList</i>
Output:	<i>org.lexgrid.valuedomain.dto.ResolvedValueDomainDefinition</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p><b>Implementation:</b></p> <p><b>Step 1:</b> Call this method on the associated LexEVS Value Domain Service instance to resolve supplied value domain against list of Coding Scheme versions if supplied. If Coding Scheme Versions list is not supplied, API will resolve against any version of the loaded Coding Scheme that is referenced by the Value Domain. Returns object <i>ResolvedValueDomainDefinition</i>, which contains <i>AbsoluteCodingSchemeVersionReferenceList</i>, which tells the Coding Scheme and the versions that were used for this resolve, plus the <i>ResolvedConceptReferencesIterator</i>, which is an iterator for all the valid concepts that belong to the Value Domain and Value Domain Definition itself.</p> <p><b>Sample Call:</b></p> <p><b>Step 1:</b> Instantiate <i>LexEVSValueDomainServices</i> if it is not done yet :  <i>org.lexgrid.valuedomain.LexEVSValueDomainServices vds = new org.lexgrid.valuedomain.impl.LexEVSValueDomainServicesImpl();</i></p> <p><b>Step 2:</b> Create <i>AbsoluteCodingSchemeVersionReferenceList</i>:  <i>AbsoluteCodingSchemeVersionReferenceList csvList = new AbsoluteCodingSchemeVersionReferenceList();</i>  <i>csvList.addAbsoluteCodingSchemeVersionReference(Constructors.createAbsoluteCodingSchemeVersionReference("Automobiles", "2.0"));</i>  <i>csvList.addAbsoluteCodingSchemeVersionReference(Constructors.createAbsoluteCodingSchemeVersionReference("AutomobilesParts", "2.0"));</i></p> <p><b>Step 3:</b> Call <i>resolveValueDomain</i> method:  <i>ResolvedValueDomainDefinition rvdDef = vds.resolveValueDomain ("valueDomainURI",csvList);</i></p>

## isSubDomain

This method checks if *childValueDomain* is a child of *parentValueDomain*.

*isSubDomain(URI childValueDomainURI, URI parentValueDomainURI)*

isSubDomain(URI childValueDomainURI, URI parentValueDomainURI)	
Description:	Checks whether <i>childValueDomainURI</i> is a child of <i>parentValueDomainURI</i> .
Input:	<i>java.net.URI</i> , <i>java.net.URI</i>
Output:	boolean
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p><b>Implementation:</b></p> <p><b>Step 1:</b> Call this method on the associated LexEVS Value Domain Service instance to determine if all the concepts that get resolved from the supplied Child Value Domain URI are children of concepts that get resolved from the Parent Value Domain URI. Returns true if they all are; otherwise, returns false.</p> <p><b>Sample Call:</b></p> <p><b>Step 1:</b> Instantiate <i>LexEVSValueDomainServices</i> if it is not done yet :  <i>org.lexgrid.valuedomain.LexEVSValueDomainServices vds = new org.lexgrid.valuedomain.impl.LexEVSValueDomainServicesImpl();</i></p> <p><b>Step 2:</b> Call <i>isSubDomain</i> method:  <i>boolean isSubDomain = vds.isSubDomain (childValueDomainURI, parentValueDomainURI);</i></p>



## getValueDomainDefinition

Utility method that returns value domain definition for the supplied value domain URI.

[getValueDomainDefinition\(Uri valueDomainURI\)](#)

getValueDomainDefinition(Uri valueDomainURI)	
Description:	Returns value domain definition for the supplied value domain URI.
Input:	<i>java.net.URI</i>
Output:	<i>org.LexGrid.emf.valueDomains.ValueDomainDefinition</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p><b>Implementation:</b></p> <p><b>Step 1:</b> Call this method on the associated LexEVS Value Domain Service instance to get the Value Domain Definition of the supplied Value Domain URI.</p> <p><b>Sample Call:</b></p> <p><b>Step 1:</b> Instantiate LexEVSVValueDomainServices if it is not done yet :</p> <pre>org.lexgrid.valuedomain.LexEVSVValueDomainServices vds = new org.lexgrid.valuedomain.impl.LexEVSVValueDomainServicesImpl();</pre> <p><b>Step 2:</b> Call getValueDomainDefinition method: <i>ValueDomainDefinition vdDef = vds.getValueDomainDefinition (valueDomainURI);</i></p>

## listValueDomains

Utility method that lists all the value domain(s) that matches supplied name.

[listValueDomains\(String valueDomainName\)](#)

listValueDomains(String valueDomainName)	
Description:	<p>Returns the URI's for the value domain definition(s) for the supplied domain name. If the name is null, returns everything. If the name is not null, returns the value domain(s) that have the assigned name.</p> <p>Note: plural because there is no guarantee of valueDomain uniqueness. If the name is the empty string "", returns all unnamed valueDomains.</p>
Input:	<i>java.lang.String</i>
Output:	<i>java.net.URI[]</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p><b>Implementation:</b></p> <p><b>Step 1:</b> Call this method on the associated LexEVS Value Domain Service instance to get the list of Value Domain URIs that matches the supplied name.</p> <p><b>Sample Call:</b></p> <p><b>Step 1:</b> Instantiate LexEVSVValueDomainServices if it is not done yet :</p> <pre>org.lexgrid.valuedomain.LexEVSVValueDomainServices vds = new org.lexgrid.valuedomain.impl.LexEVSVValueDomainServicesImpl();</pre> <p><b>Step 2:</b> Call listValueDomains method: <i>URI[] uris = vds.listValueDomains("someValueDomainName");</i></p>

## getAllValueDomainsWithNoNames

This method returns all the value domain definition URI's that contains no names.

[getAllValueDomainsWithNoNames\(\)](#)

getAllValueDomainsWithNoNames()	
Description:	Returns the URI's of all unnamed value domain definition(s).
Input:	none
Output:	<i>java.net.URI[]</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation	<b>Implementation:</b>



Details:	<p><b>Step 1:</b> Call this method on the associated LexEVS Value Domain Service instance to get the list of Value Domain URI that have no names.</p> <p><b>Sample Call:</b></p> <p><b>Step 1:</b> Instantiate LexEVSVValueDomainServices if it is not done yet :</p> <pre>org.lexgrid.valuedomain.LexEVSVValueDomainServices vds = new org.lexgrid.valuedomain.impl.LexEVSVValueDomainServicesImpl();</pre> <p><b>Step 2:</b> Call getAllValueDomainsWithNoNames method: <code>URI[] uris = vds.getAllValueDomainsWithNoNames();</code></p>
----------	--

#### getValueDomainEntitiesForTerm

This method resolves the supplied value domain and returns only those concept codes that matches the supplied term.

`getValueDomainEntitiesForTerm(String term, URI valueDomainURI, String matchAlgorithm)`

getValueDomainEntitiesForTerm(String term, URI valueDomainURI, String matchAlgorithm)	
Description:	<p>Resolves the value domain supplied and restricts it to the term and matchAlgorithm supplied. The returned object ResolvedValueDomainCodedNodeSet contains the codingScheme URI and Version that was used to resolve and the CodedNodeSet.</p> <p>Note: the CodedNodeSet is unresolved</p>
Input:	<pre>java.lang.String, java.net.URI, java.lang.String</pre>
Output:	<code>org.lexgrid.valuedomain.dto.ResolvedValueDomainCodedNodeSet</code>
Exception:	<code>org.LexGrid.LexBIG.Exceptions.LBException</code>
Implementation Details:	<p><b>Implementation:</b></p> <p><b>Step 1:</b> Call this method on the associated LexEVS Value Domain Service instance to get the resolved Value Domain Entries as CodedNodeSet that is restricted to the supplied term and the match algorithm. The returned CodedNodeSet is not resolved.</p> <p><b>Sample Call:</b></p> <p><b>Step 1:</b> Instantiate LexEVSVValueDomainServices if it is not done yet :</p> <pre>org.lexgrid.valuedomain.LexEVSVValueDomainServices vds = new org.lexgrid.valuedomain.impl.LexEVSVValueDomainServicesImpl();</pre> <p><b>Step 2:</b> Call getValueDomainEntriesForTerm method: <code>ResolvedValueDomainCodedNodeSet vdvnms = vds.getValueDomainEntriesForTerm ("General Motors", new URI("AUTO:AllDomesticANDGM"), MatchAlgorithms.exactMatch.name());</code></p>

#### getCodingSchemesInValueDomain

This method returns all the coding scheme referenced by supplied value domain.

`getCodingSchemesInValueDomain(URI valueDomainURI)`

getCodingSchemesInValueDomain(URI valueDomainURI)	
Description:	Returns list of coding scheme summary that is referenced by the supplied value domain.
Input:	<code>java.net.URI</code>
Output:	<code>org.LexGrid.LexBIG.DataModel.Collections.AbsoluteCodingSchemeVersionReferenceList</code>
Exception:	<code>org.LexGrid.LexBIG.Exceptions.LBException</code>
Implementation Details:	<p><b>Implementation:</b></p> <p><b>Step 1:</b> Call this method on the associated LexEVS Value Domain Service instance to get List of all Coding Scheme URIs and the Versions that the supplied Value Domain URI references.</p> <p><b>Sample Call:</b></p> <p><b>Step 1:</b> Instantiate LexEVSVValueDomainServices if it is not done yet :</p>

```
org.lexgrid.valuedomain.LexEVSValueDomainServices vds = new
org.lexgrid.valuedomain.impl.LexEVSValueDomainServicesImpl();
Step 2: Call getCodingSchemesInValueDomain method: AbsoluteCodingSchemeVersionReferenceList
csvList = vds.getCodingSchemesInValueDomain(new URI("AUTO:AllDomesticANDGM"));
```

isDomain

This method checks if the supplied entityCode is of type 'valueDomain' in supplied coding scheme and version.

`isDomain(String entityCode, String codingSchemeName, CodingSchemeVersionOrTag csvt)`

<code>isDomain(String entityCode, String codingSchemeName, CodingSchemeVersionOrTag csvt)</code>	
Description:	Determines if the supplied entity code is of type valueDomain in the supplied coding scheme and, if it is, returns true; otherwise returns false.
Input:	<i>java.lang.String</i> , <i>java.lang.String</i> <i>org.LexGrid.LexBIG.DataModel.Core.CodingSchemeVersionOrTag</i>
Output:	<i>boolean</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p><b>Implementation:</b></p> <p><i>Step 1:</i> Call this method on the associated LexEVS Value Domain Service instance to check if the supplied entity code is of type valueDomain in the supplied Coding Scheme.</p> <p><b>Sample Call:</b></p> <p><i>Step 1:</i> Instantiate LexEVSValueDomainServices if it is not done yet :</p> <pre>org.lexgrid.valuedomain.LexEVSValueDomainServices vds = new org.lexgrid.valuedomain.impl.LexEVSValueDomainServicesImpl();</pre> <p><i>Step 2:</i> Call isDomain method: <i>boolean isDomain = vds.isDomain ("VD005", "Automobiles", Constructors.createCodingSchemeVersionOrTag(null, "2.0"));</i></p>

## Resolved Value Domain Objects

### ResolvedValueDomainCodedNodeSet

Contains Coding Scheme Version reference list that was used to resolve the Value Domain and the CodedNodeSet. The CodedNodeSet is not resolved.

### ResolvedValueDomainDefinition

A resolved Value Domain Definition containing the Coding Scheme Version reference list that was used to resolve the Value Domain and an iterator for resolved concepts.

## Error Handling

LexEVS Value Domain services uses *org.LexGrid.LexBIG.impl.loaders.MessageDirector* to direct all fatal, error, warning, info messages with appropriate messages to the LexBIG log files in the 'log' folder of LexEVS install directory.

Along with *MessageDirector*, the services will also make use of *org.LexGrid.LexBIG.exception.LBException* to throw any fatal and error messages to the log file as well as to console.

## Load Scripts

Following are the scripts that can be used to load value domain definition that are in LexGrid XML format into the system :

- *LoadValueDomain.bat* for Windows environment and
- *LoadValueDomain.sh* for Unix environment

The above scripts can be found under 'Admin' folder of LexEVS install directory.

Both scripts take in the following parameters:

-in

Input <uri>

URI or path specifying location of the source file.

-v

Validate <int>

Perform validation of the candidate resource without loading data. Supported levels of validation include:

0 = Verify document is well-formed

1 = Verify document is valid

Example:

sh LoadValueDomain.sh -in "file:///path/to/file.xml"

## Sample Value Domain Definition XML file

Below is a sample XML file containing Value Domain Definitions in LexGrid format that can be loaded using LexEVS Value Domain Service.

```
<?xml version="1.0" encoding="UTF-8"?>
<systemRelease xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://LexGrid.org/schema/2009/01/LexGrid/versions http://LexGrid.org/schema/2009/01/LexGrid/versions.xsd"
xmlns="http://LexGrid.org/schema/2009/01/LexGrid/versions" xmlns:lgVer="http://LexGrid.org/schema/2009/01/LexGrid/versions"
xmlns:lgCommon="http://LexGrid.org/schema/2009/01/LexGrid/commonTypes" xmlns:data="data"
xmlns:lgVD="http://LexGrid.org/schema/2009/01/LexGrid/valueDomains" xmlns:lgNaming="http://LexGrid.org/schema/2009/01/LexGrid/naming"
releaseURI="http://testRelease/04" releaseDate="2008-11-07T14:55:51.615-06:00">
  <lgCommon:entityDescription>Sample value domains</lgCommon:entityDescription>
  <lgVer:valueDomains>
    <lgVD:mappings>
      <lgNaming:supportedAssociation localId="hasSubtype" uri="urn:oid:1.3.6.1.4.1.2114.108.1.8.1">hasSubtype</lgNaming:supportedAssociation>
      <lgNaming:supportedCodingScheme localId="Automobiles" uri="urn:oid:1.1.1.0.1">Automobiles</lgNaming:supportedCodingScheme>
      <lgNaming:supportedDataType localId="testhtml">test/html</lgNaming:supportedDataType>
      <lgNaming:supportedDataType localId="text/plain">text/plain</lgNaming:supportedDataType>
      <lgNaming:supportedHierarchy localId="is_a" associationNames="hasSubtype" isForwardNavigable="true" rootCode="@">hasSubtype</lgNaming:supportedHierarchy>
      <lgNaming:supportedLanguage localId="en" uri="www.en.org/orsomething">en</lgNaming:supportedLanguage>
      <lgNaming:supportedNamespace localId="Automobiles" uri="urn:oid:1.1.1.0.1" equivalentCodingScheme="Automobiles">Automobiles</lgNaming:supportedNamespace>
      <lgNaming:supportedProperty localId="textualPresentation">textualPresentation</lgNaming:supportedProperty>
      <lgNaming:supportedSource localId="lexgrid.org">lexgrid.org</lgNaming:supportedSource>
      <lgNaming:supportedSource localId="_111101">11.11.0.1</lgNaming:supportedSource>
    </lgVD:mappings>
    <lgVD:valueDomainDefinition valueDomainURI="SRITEST:AUTO:DomesticAutoMakers" valueDomainName="Domestic Auto Makers" defaultCodingScheme="Automobiles" effectiveDate="2009-01-01T11:00:00Z"
isActive="true" status="ACTIVE">
      <lgVD:properties>
        <lgCommon:property propertyName="textualPresentation">
          <lgCommon:value> Domestic Auto Makers</lgCommon:value>
        </lgCommon:property>
      </lgVD:properties>
      <lgVD:definitionEntry ruleOrder="1" operator="OR">
        <lgVD:entityReference entityCode="005" referenceAssociation="hasSubtype" transitiveClosure="true" targetToSource="false" leafOnly="false"/>
      </lgVD:definitionEntry>
    </lgVD:valueDomainDefinition>
    <lgVD:valueDomainDefinition valueDomainURI="SRITEST:AUTO:AllDomesticButGM" valueDomainName="All Domestic Autos But GM" defaultCodingScheme="Automobiles" effectiveDate="2009-01-01T11:00:00Z"
isActive="true" status="ACTIVE">
      <lgVD:properties>
        <lgCommon:property propertyName="textualPresentation">
          <lgCommon:value> Domestic Auto Makers</lgCommon:value>
        </lgCommon:property>
      </lgVD:properties>
      <lgVD:definitionEntry ruleOrder="1" operator="OR">
        <lgVD:entityReference entityCode="005" referenceAssociation="hasSubtype" transitiveClosure="true" targetToSource="false" leafOnly="false"/>
      </lgVD:definitionEntry>
      <lgVD:definitionEntry ruleOrder="2" operator="SUBTRACT">
        <lgVD:entityReference entityCode="GM" referenceAssociation="hasSubtype" transitiveClosure="true" targetToSource="false" leafOnly="false"/>
      </lgVD:definitionEntry>
    </lgVD:valueDomainDefinition>
    <lgVD:valueDomainDefinition valueDomainURI="SRITEST:AUTO:AllDomesticANDGM" valueDomainName="All Domestic Autos AND GM" defaultCodingScheme="Automobiles" effectiveDate="2009-01-01T11:00:00Z"
isActive="true" status="ACTIVE">
      <lgVD:properties>
        <lgCommon:property propertyName="textualPresentation">
          <lgCommon:value> Domestic Auto Makers AND GM</lgCommon:value>
        </lgCommon:property>
      </lgVD:properties>
      <lgVD:definitionEntry ruleOrder="1" operator="OR">
        <lgVD:entityReference entityCode="005" referenceAssociation="hasSubtype" transitiveClosure="true" targetToSource="false" leafOnly="false"/>
      </lgVD:definitionEntry>
      <lgVD:definitionEntry ruleOrder="2" operator="AND">
        <lgVD:entityReference entityCode="GM" referenceAssociation="hasSubtype" transitiveClosure="true" targetToSource="false" leafOnly="false"/>
      </lgVD:definitionEntry>
    </lgVD:valueDomainDefinition>
    <lgVD:valueDomainDefinition valueDomainURI="SRITEST:AUTO:DomesticLeafOnly" valueDomainName="Domestic Leaf Only" defaultCodingScheme="Automobiles" effectiveDate="2009-01-01T11:00:00Z"
isActive="true" status="ACTIVE">
      <lgVD:properties>
        <lgCommon:property propertyName="textualPresentation">
          <lgCommon:value>Domestic Leaf Only</lgCommon:value>
        </lgCommon:property>
      </lgVD:properties>
      <lgVD:definitionEntry ruleOrder="1" operator="OR">
        <lgVD:entityReference entityCode="005" referenceAssociation="hasSubtype" transitiveClosure="true" targetToSource="false" leafOnly="true"/>
      </lgVD:definitionEntry>
    </lgVD:valueDomainDefinition>
  </lgVer:valueDomains>
</systemRelease>
```

## System Testing

The System test case for the LexEVS Value Domain service is performed using the JUnit test suite:

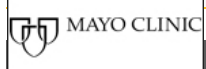
*org.LexGrid.LexBIG.Impl.testUtility.VDAAllTests*

This test suite will be run as part of regular LexEVS test suites AllTestsAllConfigs and AllTestsNormalConfigs.

## Installation / Packaging

Value Domain Services are integrated parts of core LexEVS API and are packaged and installed with other LexEVS services.

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



This page was last modified on 22 December 2009, at 13:01. This page has been accessed 148 times.

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

