

LexEVS 6.x Value Set Service

Contents of this Page

- [Introduction](#)
- [LexEVSVValueSetDefinitionServices Class Diagram](#)
- [LexEVS Value Set Definition Services API](#)
 - [Administration Functions](#)
 - [Loading Value Set Definitions](#)
 - [Loading Value Set Definition Object](#)
 - [Loading Value Set Definitions in File](#)
 - [Export Value Set](#)
 - [Export Value Set Definition](#)
 - [Export Value Set Resolution](#)
 - [Remove Value Set Definition](#)
 - [Query Functions](#)
 - [Validate XML Resources](#)
 - [isEntityInValueSet](#)
 - [isEntityInValueSet\(String entityCode, URI entityCodeNamespace, URI valueSetDefinitionURI, String valueSetDefinitionRevisionId, AbsoluteCodingSchemeVersionReferenceList csVersionList, String versionTag\)](#)
 - [listValueSetsWithEntityCode](#)
 - [getCodedNodeSetForValueSetDefinition](#)
 - [isSubSet](#)
 - [getValueSetDefinition](#)
 - [listValueSetDefinitions](#)
 - [listValueSetDefinitionURIs](#)
 - [getAllValueSetDefinitionsWithNoNames](#)
 - [getValueSetDefinitionEntitiesForTerm](#)
 - [getCodingSchemesInValueSetDefinition](#)
 - [isValueSetDefinition](#)
 - [getValueSetDefinitionURIsWithCodingScheme](#)
 - [getValueSetDefinitionURIsWithConceptDomain](#)
 - [getValueSetDefinitionURIsWithUsageContext](#)
 - [getValueSetDefinitionURIsWithConceptDomainAndUsageContext](#)
 - [Resolve Functions](#)
 - [Resolving Stored Value Set Definition](#)
 - [Resolving Supplied Value Set Definition Objects](#)
 - [Resolved Value Set Objects](#)
 - [ResolvedValueSetCodedNodeSet](#)
 - [ResolvedValueSetDefinition](#)
 - [Error Handling](#)
 - [Load Scripts](#)
 - [Sample Value Set Definition XML File](#)
 - [System Testing](#)
 - [Installation / Packaging](#)
- [Value Set GUI](#)

LexEVS Value Set Links

- [Value Set Guide Main Page](#)
 - [Value Set Design](#)
 - [Value Set Service API](#)
 - [Pick List Design](#)
 - [Pick List Service API](#)
 - [Value Set GUI](#)
- [Programmer's Guide Main Page](#)
- [LexEVS 6.0 Main Page](#)
- [LexEVS Current Release](#)

Introduction

This document is a section of the [LexEVS 6.0 Programmer's Guide](#).

The Value Set services are integrated parts of the LexEVS API. It provide three major functions:

- **Administration**- Ability to load, export and remove value set definitions
 - **Loader** - Ability to load Value Set Definitions programmatically into the LexGrid repository using the domain objects that are available via the LexGrid logical model
 - **Exporter** - Ability to export Value Set Definition and Value Set Resolution to a file in LexGrid XML format
 - **Remove** - Ability to remove Value Set Definition from the system

**Note**

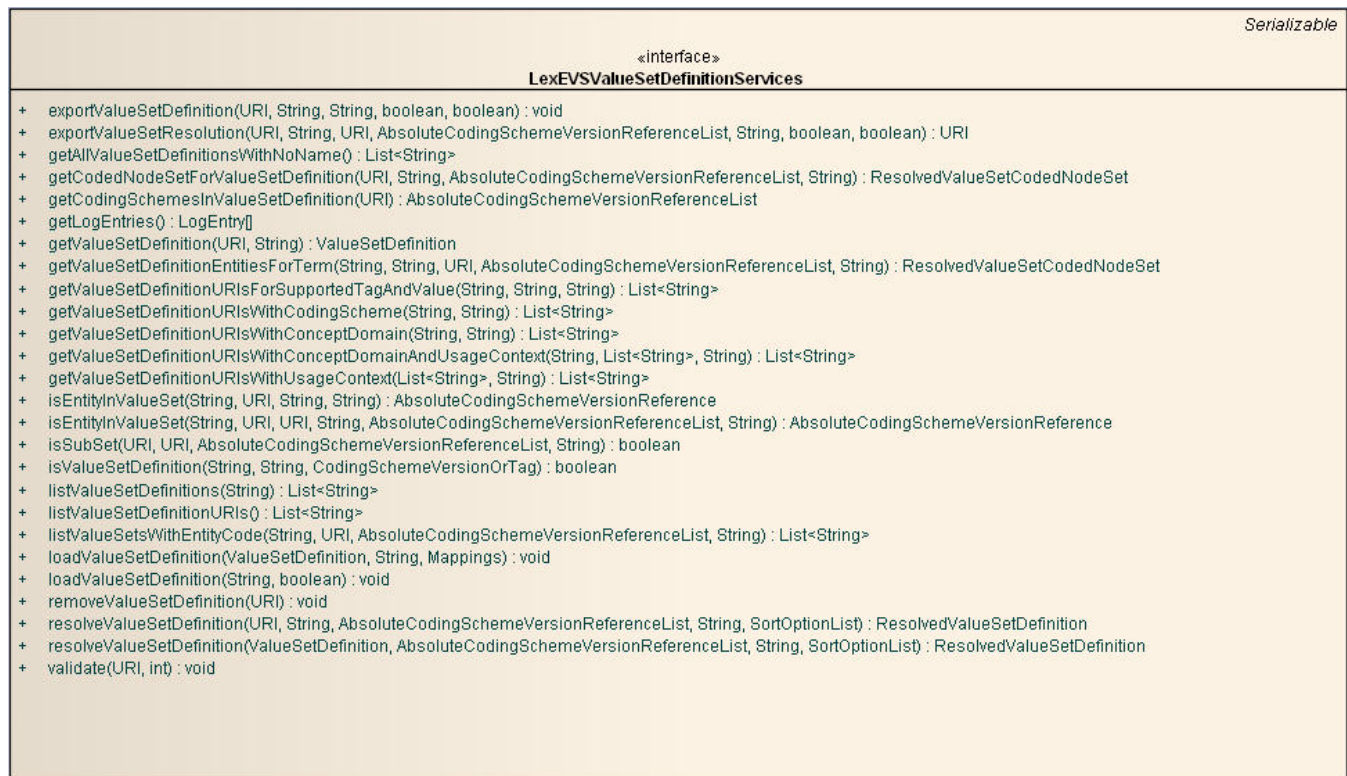
These administration operations can be performed in LexEVS 'Local' environment only. This can not be performed using LexEVS Distributed environment.

- **Query** - Ability to apply user restrictions (ex: valueSet URI) and dynamically resolve the definitions at run time
- **Resolve** - Ability to resolve Value Set Definition dynamically against selected Coding Scheme Version(s) and return all the concepts belonging to the value set

The LexEVS Value Set Services expose the API particularly for the Value Set Definition elements of the LexGrid Logical Model. For more information on LexGrid model see the [LexBig Model and Schema](#)

LexEVSVValueSetDefinitionServices Class Diagram

LexEVSVValueSetDefinitionServices is the main interface for all the services provided by LexEVS Value Set API. Here is the class diagram of LexEVSVValueSetDefinitionServices:



LexEVS Value Set Definition Services API

LexEVS Value Set Definition Services provides three major functions:

- Administration function
- Query function and
- Resolve function

Each of these functions are described in following sections.

Administration Functions

LexEVS Value Set Definition Services provide s following administration functions:

- Load
- Export and
- Remove value set definitions.



Note

These administration operations can be performed in LexEVS 'Local' environment only. This can not be performed using LexEVS Distributed environment.

Loading Value Set Definitions

There are two methods that could be used to load Value Set Definitions:

- Loading Value Set Definition Object - This function provides the capability to load supplied ValueSetDefinition object into the system.
- Loading Value Set Definitions in file - This function provides the capability to load Value Set Definitions found in file that are in LexGrid XML format.

Loading Value Set Definition Object

This function provides the capability to load supplied ValueSetDefinition object into the system.

```
loadSetValueSetDefinition(ValueSetDefinition vsddef, String systemReleaseURI, Mappings mappings)
```

Description:	Loads supplied valueSetDefinition object. Optionally, it can assign the release URI and mappings (supported attributes like SupportedCodingScheme, SupportedConceptDomain, SupportedAssociation etc) to the valueSetDefinition being loaded.
Input:	<i>org.LexGrid.valueSets.ValueSetDefinition</i> vsdDef - (Mandatory) Value Set Definition object to be loaded. <i>java.lang.String</i> systemReleaseURI - (Optional) System Release URI this Value Set Definition belongs to. <i>org.LexGrid.naming.Mappings</i> mappings - (Optional) List of Supported Attributes this Value Set Definition uses.
Output:	none
Exception:	<i>LBException</i>
Implementation Details:	<p>Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Value Set Definition Service instance to load a Value Set Definition object and optionally assign the System Release URI and mappings to this definition. Sample Call:</p> <ul style="list-style-type: none">• <i>Step 1:</i> Instantiate LexEVSVValueSetDefinitionServices if it is not done yet: <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> • <i>Step 2:</i> Create and populate the ValueSetDefinition object. ValueSetDefinition object can be created using: <pre>org.LexGrid.valueSets.ValueSetDefinition vsd = new org.LexGrid.valueSets. ValueSetDefinition;</pre> <p>Data for ValueSetDefinition object can be populated by using set methods:</p> <pre>vsd.setValueSetDefinitionURI(uri); vsd.setValueSetDefinitionName(name); vsd.setDefaultCodingScheme(cs); vsd.setEntityDescription(ed);</pre> <p>Similarly, DefinitionEntry, Property, and Mapping objects can be created and assigned to the valueSetDefinition object:</p> <pre>vsd.getDefinitionEntry.add(vdEntry); vsd.setProperties(propertisObject); vsd.setMappings(mappingsObject);</pre> <ul style="list-style-type: none">• <i>Step 3:</i> Call the load method by passing the Value Set Definition object and optionally the System Release URI and mappings: <pre>vsdServ.loadValueSetDefinition(vsd, systemReleaseURI, mappings);</pre>

Loading Value Set Definitions in File

This function provides the capability to load Value Set Definitions found in file that are in LexGrid XML format.

```
loadValueSetDefinition(String xmlFileLocation, boolean failOnAllErrors)
```

Description:	Loads valueSetDefinitions found in input XML file. The input XML file should in LexGrid format i.e. the contents should confirm to LexGrid valueSets schema [
Input:	<i>java.lang.String</i> xmlFileLocation - (Mandatory) File containing ValueSetDefinitions in LexGrid XML format. <i>boolean</i> failOnErrors - (Optional) default is false.
Output:	none
Exception:	<i>LBException</i>
Implementation Details:	<p>Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Value Set Definition Service instance to load all Value Set Definitions found in an XML file that is in LexGrid format. Sample Call:</p> <ul style="list-style-type: none"><i>Step 1:</i> Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"><i>Step 2:</i> Call load method by passing the inputfile location and boolean flag whether to stop on load errors: <pre>vsdServ.loadvalueSetDefinition(inputXMLFile, true);</pre>

Export Value Set

There are two export functions available :

1. Export Value Set Definition - This function provides the capability to export Value Set Definition to a file in LexGrid XML format. This will be helpful if there is a need to import this exported Value Set Definition in different instance of LexEVS.
2. Export Value Set Resolution (Expanded Value Set) - This function provides the capability to resolve the Value Set Definition and exports the resolved (expanded) contents as a LexGrid Coding Scheme in XML format.

Export Value Set Definition

This function provides the capability to export Value Set Definition to a file in LexGrid XML format. This will be helpful if there is a need to import this exported Value Set Definition in different instance of LexEVS.

```
exportValueSetDefinition(URI valueSetDefURI, String valueSetDefinitionRevisionId, String xmlFullPathName, boolean  
overwrite, boolean failOnAllErrors)
```

Description:	Export Value Set Definition to a file in LexGrid XML format.
Input:	<i>java.net.URI</i> valueSetDefinitionURI - (Mandatory) the URI of value set definition to be exported. <i>java.lang.String</i> valueSetDefinitionRevisionId - (Optional) revision id of the value set definition to export <i>java.lang.String</i> xmlFullPathName - (Mandatory) Location to save the definition <pre>boolean overwrite - True: to override the existing file. boolean failOnAllErrors - True: stops exporting if any error. </pre>
Output:	none
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to export the Value Set Definition to a file in LexGrid XML format. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Call exportValueSetDefinition method and provide parameter values : <pre>vsdServ.exportValueSetDefinition(valueSetDefinitionURI, null, true, false);</pre>
--------------------------------	---

Export Value Set Resolution

This function provides the capability to export Value Set Resolution (Expanded Value Set) to a file in LexGrid XML format as a Coding Scheme. This will be helpful if there is a need to import the contents of a value set as a separate Coding Scheme. The URI of the Coding Scheme will be set to URI of Value Set Definition being exported. And all the contents of the value set will be represented as an entities within that Coding Scheme.

```
exportValueSetResolution(Uri valueSetDefinitionURI, String valueSetDefinitionRevisionId, Uri exportDestination,
AbsoluteCodingSchemeVersionReferenceList csVersionList, String csVersionTag, boolean overwrite, boolean
failOnAllErrors)
```

Description:	Export Value Set Resolution to a file in LexGrid XML format as a Coding Scheme.
Input:	<p><i>java.net.Uri</i> valueSetDefinitionURI - (Mandatory) the URI of value set definition to be exported. <i>java.lang.String</i> valueSetDefinitionRevisionId - (Optional) revision id of the value set definition to export <i>java.net.Uri</i> exportDestination - (Mandatory) Location to save the definition <i>org.LexGrid.LexBIG.DataModel.Collections.AbsoluteCodingSchemeVersionReferenceList</i> csVersionList - A list of coding scheme URI's and versions to be used. These will be used only if they are present in the service. If absent, the most recent version will be used instead. boolean overwrite - True: to override the existing file. boolean failOnAllErrors - True: stops exporting if any error.</p>
Output:	none
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to export the Value Set Resolution to a file in LexGrid XML format as Coding Scheme. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Call exportValueSetResolution method and provide parameter values : <pre>String codingSchemeName = "NCI_Thesaurus"; String version = "10.08e"; AbsoluteCodingSchemeVersionReferenceList csVersionList = new AbsoluteCodingSchemeVersionReferenceList(); csVersionList.addAbsoluteCodingSchemeVersionReference(Constructors. createAbsoluteCodingSchemeVersionReference(codingSchemeName, version)); vsdServ.exportValueSetResolution(valueSetDefinitionURI, null, csVersionList, "PRODUCTION", true, false);</pre>

Remove Value Set Definition

Below is the function to remove value set definition from the system.

```
removeValueSetDefinition(Uri valueSetDefinitionURI)
```

Description:	Removes supplied value set definition from the system.
Input:	<i>java.net.URI</i> valueSetDefinitionURI - (Mandatory) the URI of value set definition to be removed.
Output:	none
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to remove the Value Set Definition from the system that matches the supplied URI. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> Step 2: Call removeValueSetDefinition method <pre>vsdServ.removeValueSetDefinition(valueSetDefinitionURI);</pre>

Query Functions

Here are some of the query functions that can be run against value set definitions using the LexEVS Value Set Definition Services.

Validate XML Resources

Function to perform validation of source XML file without loading.

```
validate(URI xmlFileLocation, int valicationLevel)
```

Description:	Performs validation of the candidate resource without loading data.
Input:	<p><i>java.net URI</i> xmlFileLocation - (Mandatory) File containing ValueSetDefinitions in LexGrid XML format. int validationLevel - validation level includes :</p> <ul style="list-style-type: none"> 0 = Verify document is well-formed 1 = Verify document is valid
Output:	none
Exception:	<i>Org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to validate the XML file that is in LexGrid format. This call will not load the data in XML file. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> Step 2: Call validate method for validation by supplying URI of the XML file and validation level. <pre>java.net URI xmlFileLocation = new URI("C:\\myVSDFile.xml"); vsdServ.validaten(xmlFileLocation, true);</pre>

isEntityInValueSet

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

```
isEntityInValueSet(String entityCode, URI valueSetDefinitionURI, String valueSetDefinitionRevisionId, String versionTag)
```

Description:	Determine whether the supplied entity code is a valid entity code somewhere in the supplied value set definition. This function is intended for use with simple value set definition that are drawn from a single coding scheme where most parameters can be defaulted.
Input:	<i>java.lang.String</i> entityCode - (Mandatory) the entity code to search for <i>java.net.URI</i> valueSetDefinitionURI - (Mandatory) the URI of the value set definition to search <i>java.lang.String</i> valueSetDefinitionRevisionId - (Optional) the version of the value set definition <i>java.lang.String</i> versionTag - (Optional) the version or tag (e.g. "devel", "production", etc.) to be used for <i>all</i> of the coding schemes searched.
Output:	<i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference</i> - coding scheme and version if the entityCode is valid, null otherwise
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to determine if the supplied entity code is a valid in the supplied Value Set Definition URI. If it is, returns Coding Scheme URI and the Version that was used to resolve; otherwise, returns null. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> Step 2: Call isEntityInValueSet method: <pre>java.net.URI valueSetDefURI = new URI("http://someDomain.html/ValueSets/NCIT/C3367_Value_set"); String entityCode = "C3367"; AbsoluteCodingSchemeVersionReference acsvr = vsdServ.isEntityInValueSet(entityCode, valueSetDefURI, null, "PRODUCTION");</pre>

```
isEntityInValueSet(String entityCode, URI entityCodeNamespace, URI valueSetDefinitionURI, String valueSetDefinitionRevisionId, AbsoluteCodingSchemeVersionReferenceList csVersionList, String versionTag)
```

```
isEntityInValueSet(String entityCode, URI entityCodeNamespace, URI valueSetDefinitionURI, String valueSetDefinitionRevisionId, AbsoluteCodingSchemeVersionReferenceList csVersionList, String versionTag)
```

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 set definition when resolved against the supplied Coding Scheme Version or Tag.
Input:	<i>java.lang.String</i> entityCode - (Mandatory) the entity code to search for <i>java.net.URI</i> entityCodeNamespace - (Optional) the URI of the entity code namespace. If omitted, the default coding scheme namespace for the value set definition will be used, if it is present. Otherwise the first matching entity code, if any, will pass <i>java.net.URI</i> valueSetDefinitionURI - (Mandatory) the URI of the value set definition to search <i>java.lang.String</i> valueSetDefinitionRevisionId - (Optional) the version of the value set definition <i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference</i> csVersionList - (Optional) a list of coding scheme URI's and versions to be used. These will be used only if they are present in the service. If absent, the most recent version will be used instead. <i>java.lang.String</i> versionTag - (Optional) the version or tag (e.g. "devel", "production", etc.) to be used for <i>all</i> of the coding schemes searched.
Output:	<i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference</i> - The codingScheme URI and version of that asserts that the code is in the value set
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to determine if the supplied entity code and its entity namespace is valid in the supplied Value Set Definition 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>Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Call isEntityInValueSet method: <pre>java.net.URI valueSetDefURI = new URI("http://someDomain.html/ValueSets/NCIT /C3367_Value_set"); String codingSchemeName = "NCI_Thesaurus"; String version = "10.08e"; String entityCode = "C3367"; String entityCodeNamespace = "NCI_Thesaurus"; AbsoluteCodingSchemeVersionReferenceList csVersionList = new AbsoluteCodingSchemeVersionReferenceList(); csVersionList.addAbsoluteCodingSchemeVersionReference(Constructors. createAbsoluteCodingSchemeVersionReference(codingSchemeName, version)); AbsoluteCodingSchemeVersionReference acsvr = vsdServ.isEntityInValueSet(entityCode, entityCodeNamespace, valueSetDefURI, null, csVersionList, "PRODUCTION");</pre>
--------------------------------	--

listValueSetsWithEntityCode

Returns all the value set definition uris that contains supplied entity code.

```
listValueSetsWithEntityCode(String entityCode, URI entityCodeNamespace, AbsoluteCodingSchemeVersionReferenceList
csVersionList, String versionTag)
```

Description:	Returns all the value set definition uris that contains supplied entity code.
Input:	<i>java.lang.String</i> entityCode - (Mandatory) the entity code to search for <i>java.net.URI</i> entityCodeNamespace - (Optional)the URI of the entity code namespace. If omitted, the default coding scheme namespace for the value set definition will be used, if it is present. Otherwise the first matching entity code, if any, will pass <i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference</i> csVersionList - (Optional) a list of coding scheme URI's and versions to be used. These will be used only if they are present in the service. If absent, the most recent version will be used instead. <i>java.lang.String</i> versionTag - (Optional) the version or tag (e.g. "dev", "production", etc.) to be used for <i>all</i> of the coding schemes searched.
Output:	<i>java.util.List<String></i> - list of value set definition URIs that contain supplied entity code
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get all the value set definition uris that contain the the supplied entity code. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSValueSetDefinitionServices vsdServ = LexEVSValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Call listValueSetsWithEntityCode method: <pre>String codingSchemeName = "NCI_Thesaurus"; String version = "10.08e"; String entityCode = "C3367"; String entityCodeNamespace = "NCI_Thesaurus"; AbsoluteCodingSchemeVersionReferenceList csVersionList = new AbsoluteCodingSchemeVersionReferenceList(); csVersionList.addAbsoluteCodingSchemeVersionReference(Constructors. createAbsoluteCodingSchemeVersionReference(codingSchemeName, version)); AbsoluteCodingSchemeVersionReference acsvr = vsdServ.listValueSetsWithEntityCode (entityCode, entityCodeNamespace, csVersionList, "PRODUCTION");</pre>
--------------------------------	--

getCodedNodeSetForValueSetDefinition

Returns unresolved CodedNodeSet populated using definition entries in the value set definition.

```
getCodedNodeSetForValueSetDefinition(URI valueSetDefinitionURI, String valueSetDefinitionRevisionId,
AbsoluteCodingSchemeVersionReferenceList csVersionList, String csVersionTag)
```

Description:	Returns unresolved CodedNodeSet populated using definition entries in the value set definition.
Input:	<p><i>java.net.URI</i> valueSetDefinitionURI - (Mandatory) the URI of the value set definition <i>java.lang.String</i> valueSetDefintionRevisionId - (Optional) the version of the value set definition <i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference</i> csVersionList - (Optional) a list of coding scheme URI's and versions to be used. These will be used only if they are present in the service. If absent, the most recent version will be used instead. <i>java.lang.String</i> versionTag - (Optional) the version or tag (e.g. "devel", "production", etc.) to be used for <i>all</i> of the coding schemes searched.</p>
Output:	<i>org.lexgrid.valuesets.dto.ResolvedValueSetCodedNodeSet</i> - Unresolved CodedNodeSet populated using definition entries in the value set definition.
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get unresolved CodedNodeSet populated using definition entries in the value set definition Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Call getCodedNodeSetForValueSetDefinition method: <pre>java.net.URI valueSetDefURI = new URI("http://someDomain.html/ValueSets/NCIT /C3367_Value_set"); String codingSchemeName = "NCI_Thesaurus"; String version = "10.08e"; AbsoluteCodingSchemeVersionReferenceList csVersionList = new AbsoluteCodingSchemeVersionReferenceList(); csVersionList.addAbsoluteCodingSchemeVersionReference(Constructors. createAbsoluteCodingSchemeVersionReference(codingSchemeName, version)); ResolvedValueSetCodedNodeSet rvsCNS = vsdServ.getCodedNodeSetForValueSetDefinition (valueSetDefURI, null, csVersionList, "PRODUCTION");</pre>
--------------------------------	---

isSubSet

This method checks if childValueSetDefinition is a sub set of parentValueSetDefinition.

```
isSubSet(URI childValueSetDefinitionURI, URI parentValueSetDefinitionURI,
AbsoluteCodingSchemeVersionReferenceList csVersionList, String versionTag)
```

Description:	Checks whether childValueSetDefinition is a sub set of parentValueSetDefinition
Input:	<i>java.net.URI</i> childValueSetDefinitionURI - (Mandatory) child value set definition URI <i>java.net.URI</i> parentValueSetDefinitionURI- (Mandatory) parent value set definition URI <i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference</i> csVersionList - (Optional) a list of coding scheme URI's and versions to be used. These will be used only if they are present in the service. If absent, the most recent version will be used instead. <i>java.lang.String</i> versionTag - (Optional) the version or tag (e.g. "devel", "production", etc.) to be used for <i>all</i> of the coding schemes searched.
Output:	<i>boolean</i> - YES, if all the elements of the child value set are in the parent value set. NO otherwise.
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to determine if all the concepts that get resolved from the supplied Child Value Set Definition URI are children of concepts that get resolved from the Parent Value Set Definition URI. Returns true if they all are; otherwise, returns false. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Call isSubSet method: <pre>java.net.URI childValueSetDefinitionURI = new URI("http://someDomain.html/ValueSets/NCIT/Child_Value_set"); java.net.URI parentValueSetDefinitionURI = new URI("http://someDomain.html/ValueSets/NCIT/Parent_Value_set"); String codingSchemeName = "NCI_Thesaurus"; String version = "10.08e"; AbsoluteCodingSchemeVersionReferenceList csVersionList = new AbsoluteCodingSchemeVersionReferenceList(); csVersionList.addAbsoluteCodingSchemeVersionReference(Constructors. createAbsoluteCodingSchemeVersionReference(codingSchemeName, version)); boolean isSubSet = vsdServ.isSubSet(childValueSetDefinitionURI, parentValueSetDefinitionURI, csVersionList, "PRODUCTION");</pre>
--------------------------------	--

getValueSetDefinition

Utility method that returns value set definition object for the supplied value set URI.


getValueSetDefinition(URI valueSetDefinitionURI, String valueSetDefinitionRevisionId)

Description:	Returns value set definition object for the supplied value set definition URI.
Input:	<i>java.net.URI</i> valueSetDefinitionURI - (Mandatory) value set definition URI <i>java.lang.String</i> valueSetDefinitionRevisionId - (Optional) the version of the value set definition
Output:	<i>org.LexGrid.valuesets.ValueSetDefinition</i> - value set definition object
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get the Value Set Definition object of the supplied Value Set Definition URI. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Call getValueSetDefinition method: <pre>org.LexGrid.valuesets.ValueSetDefinition vsDef = vsdServ.getValueSetDefinition (valueSetDefinitionURI, null);</pre>

listValueSetDefinitions

Utility method that lists all the URI's of the value set definition(s) for the supplied value set definition name. If the name is null, returns everything. If the name is not null, returns the value set definition(s) that have the assigned name.

listValueSetDefinitions(String valueSetDefinitionName)

Description:	<p>Returns the URI's of the value set definition(s) for the supplied name. If the name is null, returns everything. If the name is not null, returns the value set definition(s) that have the assigned name.</p> <div>  Note Plural because there is no guarantee of Value Set Definition name uniqueness. If the name is the empty string "", returns all unnamed valueSetDefinition(s). </div>
Input:	<i>java.lang.String</i> valueSetDefinitionName - (Optional) name of the value set definition. If null, returns URIs all the value set definition(s) in the system, if empty string "", returns all unnamed value set definition URIs. If name is provided, returns value set definition uris with the name that matches supplied valueSetDefinitionName.
Output:	<i>java.util.List<String></i> - List of matching value set definition uris
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get the list of Value Set Definition URIs that matches the supplied name. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> Step 2: Call listValueSetDefinitions method: <pre>List<String> uris = vsdServ.listValueSetDefinitions("someValueSetDefinitionName");</pre>

listValueSetDefinitionURIs

Utility method that lists all the URI's of the value set definition(s) that are loaded in the system.

listValueSetDefinitionURIs()

Description:	Returns all the URI's of the value set definition(s) that are loaded in the system.
Input:	none
Output:	<i>java.util.List<String></i> - List of all the value set definition uris available in the system
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get all the Value Set Definition URIs available in the system. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> Step 2: Call listValueSetDefinitionURIs method: <pre>List<String> uris = vsdServ.listValueSetDefinitionURIs();</pre>

getAllValueSetDefinitionsWithNoNames

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


getAllValueSetDefinitionsWithNoNames()

Description:	Returns the URI's of all unnamed value set definition(s).
Input:	none
Output:	<i>java.util.List<String></i> - List of all unnamed value set definitions URI's
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Value Set Definition Service instance to get the list of Value Set Definition URI's that have no names. Sample Call:</p> <ol style="list-style-type: none"><i>Step 1:</i> Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"><i>Step 2:</i> Call listValueSetDefinitionURIs method: <pre>List<String> uris = vsdServ.getAllValueSetDefinitionsWithNoNames();</pre>

getValueSetDefinitionEntitiesForTerm

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

getValueSetDefinitionEntitiesForTerm(String term, String matchAlgorithm, URI valueSetDefinitionURI, AbsoluteCodingSchemeVersionReferenceList csVersionList, String versionTag)

Description:	<p>Resolves the value set definition supplied and restricts it to the term and matchAlgorithm supplied. The returned object ResolvedValueSetCodedNodeSet contains the codingScheme URI and Version that was used to resolve and the CodedNodeSet.</p> <div> Note The CodedNodeSet is unresolved.</div>
Input:	<p><i>java.lang.String</i> term - (Mandatory) text to match. Format is specific to the match algorithm <i>java.lang.String</i> matchAlgorithm - (Optional) match algorithm to use. Must be the name of a supported match algorithm. Default - Lucene Query <i>java.net.URI</i> valueSetDefinitionURI - (Mandatory) value set definition to resolve <i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference</i> csVersionList - (Optional) a list of coding scheme URI's and versions to be used. These will be used only if they are present in the service. If absent, the most recent version will be used instead. <i>java.lang.String</i> versionTag - (Optional) the version or tag (e.g. "dev", "production", etc.) to be used for <i>all</i> of the coding schemes searched.</p>
Output:	<i>org.lexgrid.valuesets.dto.ResolvedValueSetCodedNodeSet</i> - contains the codingScheme URI and Version that was used to resolve and the unresolved CodedNodeSet.
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get the resolved Value Set Entries as CodedNodeSet that is restricted to the supplied term and the match algorithm. The returned CodedNodeSet is not resolved. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Call getValueSetDefinitionEntitiesForTerm method: <pre>String codingSchemeName = "Automobiles"; String version = "1.1"; AbsoluteCodingSchemeVersionReferenceList csVersionList = new AbsoluteCodingSchemeVersionReferenceList(); csVersionList.addAbsoluteCodingSchemeVersionReference(Constructors. createAbsoluteCodingSchemeVersionReference(codingSchemeName, version)); ResolvedValueSetCodedNodeSet vsvns = vsdServ.getValueSetDefinitionEntitiesForTerm ("General Motors", MatchAlgorithms.exactMatch.name(), new URI("AUTO:AllDomesticANDGM"), csVersionList, "PRODUCTION");</pre>
--------------------------------	---

getCodingSchemesInValueSetDefinition

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

getCodingSchemesInValueSetDefinition([URI](#) valueSetDefinitionURI)

Description:	Returns list of coding scheme summary that is referenced by the supplied value set definition.
Input:	<i>java.net.URI</i> valueSetDefinitionURI - (Mandatory) value set definition uri
Output:	<i>org.LexGrid.LexBIG.DataModel.Collections.AbsoluteCodingSchemeVersionReferenceList</i> - list of coding scheme summary referenced by the supplied value set definition
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get List of all Coding Scheme URIs and the Versions that the supplied Value Set Definition URI references. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Call getCodingSchemesInValueSetDefinition method: <pre>AbsoluteCodingSchemeVersionReferenceList csvList = vsdServ. getCodingSchemesInValueSetDefinition(new URI("AUTO:AllDomesticANDGM"));</pre>

isValueSetDefinition

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

isValueSetDefinition([String](#) entityCode, [String](#) codingSchemeName, [CodingSchemeVersionOrTag](#) csvt)

Description:	Determines if the supplied entity code is of type valueSetDefinition in the supplied coding scheme and, if it is, returns true; otherwise returns false.
---------------------	--

Input:	<i>java.lang.String</i> entityCode - (Mandatory) entity code <i>java.lang.String</i> codingSchemeName - (Mandatory) name of coding scheme <i>org.LexGrid.LexBIG.DataModel.Core.CodingSchemeVersionOrTag</i> csvt - (Optional) coding scheme version or tag
Output:	<i>boolean</i> - TRUE : If entityCode is of type valueSetDefinition in supplied coding scheme, FALSE : otherwise
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to check if the supplied entity code is of type valueSetDefinition in the supplied Coding Scheme. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> Step 2: Call isValueSetDefinition method: <pre>boolean isVSD = vsdServ.isValueSetDefinition("VSD005", "Automobiles", Constructors. createCodingSchemeVersionOrTag(null, "1.1"));</pre>

getValueSetDefinitionURIsWithCodingScheme

This method returns a list of Value Set Definition URIs that references supplied coding scheme.

```
getValueSetDefinitionURIsWithCodingScheme(String codingSchemename, String codingSchemeURI)
```

Description:	Returns a list of Value Set Definition URIs that references supplied coding scheme.
Input:	<i>java.lang.String</i> codingSchemename - (Mandatory) Name of coding scheme to check <i>java.lang.String</i> codingSchemeURI - (Mandatory) URI of coding scheme that codingSchemename belongs to
Output:	<i>java.util.List<String></i> - List of value set definition URIs that references the supplied coding scheme name
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get list of value set definition uris that references the supplied coding scheme. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> Step 2: Call getValueSetDefinitionURIsWithCodingScheme method: <pre>List<String> uris = vsdServ.getValueSetDefinitionURIsWithCodingScheme("Automobiles", "urn:oid:1.11.0.1");</pre>

getValueSetDefinitionURIsWithConceptDomain

This method returns a list of Value Set Definition URIs that are bound to supplied concept domain.

```
{{getValueSetDefinitionURIsWithConceptDomain(String conceptDomain, String codingSchemeURI}}
```

Description:	Returns a list of Value Set Definition URIs that are bound to supplied concept domain.
Input:	<i>java.lang.String</i> conceptDomain- (Mandatory) name or uri of concept domain to check <i>java.lang.String</i> codingSchemeURI - (Mandatory) URI of coding scheme that concept domain belongs to

Output:	<i>java.util.List<String></i> - List of value set definition URIs that are bound to supplied concept domain
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get list of value set definition uris that are bound to the supplied concept domain. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> Step 2: Call getValueSetDefinitionURIsWithConceptDomain method: <pre>List<String> uris = vsdServ.getValueSetDefinitionURIsWithConceptDomain("Autos", "urn: oid:11.11.0.1");</pre>

getValueSetDefinitionURIsWithUsageContext

This method returns a list of Value Set Definition URIs that references supplied list of usage context(s).

```
getValueSetDefinitionURIsWithUsageContext(List<String> usageContexts, String codingSchemeURI)
```

Description:	Returns a list of Value Set Definition URIs that references supplied list of usage context(s).
Input:	<i>java.util.List<java.lang.String></i> usageContexts- (Mandatory) list of name or uri of usage context to check <i>java.lang.String</i> codingSchemeURI - (Mandatory) URI of coding scheme that usage context belongs to
Output:	<i>java.util.List<String></i> - List of value set definition URIs that references the supplied list of usage context
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get list of value set definition uris that references the supplied list of usage context. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> Step 2: Call getValueSetDefinitionURIsWithUsageContext method: <pre>List<String> usageContexts = new ArrayList<String>(); usageContexts.add("Automobiles"); usageContexts.add("Domestic Auto Makers"); List<String> uris = vsdServ.getValueSetDefinitionURIsWithUsageContext(usageContexts, "urn:oid:11.11.0.1");</pre>

getValueSetDefinitionURIsWithConceptDomainAndUsageContext

This method Returns list of Value Set Definition URIs that are bound to supplied concept domain and in supplied usage context.

```
getValueSetDefinitionURIsWithConceptDomainAndUsageContext(String conceptDomain, List<String> usageContexts,
String codingSchemeURI)
```

Description:	Returns list of Value Set Definition URIs that are bound to supplied concept domain and in supplied usage context.
Input:	<i>java.lang.String</i> conceptDomain - (Mandatory) name or uri of concept domain to check <i>java.util.List<java.lang.String></i> usageContexts- (Mandatory) list of name or uri of usage context to check <i>java.lang.String</i> codingSchemeURI - (Mandatory) URI of coding scheme that usage context and concept domain belongs to

Output:	<i>java.util.List<String></i> - List of value set definition URIs that are bound to supplied concept domain and references the supplied list of usage context
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to get list of value set definition uris that is bound to supplied concept domain and references the supplied list of usage context. Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>org.lexgrid.valuesets.LexEVSVValueSetDefinitionServices vsdServ = LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Call <code>getValueSetDefinitionURIsWithConceptDomainAndUsageContext</code> method: <pre>List<String> usageContexts = new ArrayList<String>(); usageContexts.add("Automobiles"); usageContexts.add("Domestic Auto Makers"); List<String> uris = vsdServ.getValueSetDefinitionURIsWithConceptDomainAndUsageContext ("Autos", usageContexts, "urn:oid:1.11.0.1");</pre>

Resolve Functions

Resolve functions provides the capability to resolve the value set definition against specific coding scheme version and get back the list of entities that belongs to the value set. If no coding scheme version is provided, the API will pick the latest version of the coding scheme.

There are two separate functions available to resolve value set definition :

- Resolve Stored Value Set Definition - This function allows you to resolve value set definition that are loaded in the system.
- Resolve Supplied Value Set Definition - This function allows you to pass a ValueSetDefinition object that may not be loaded in the system and get it resolved.

Resolving Stored Value Set Definition

This function resolves the value set definition that is loaded in the system and returns set of valied concept codes plus the code systems and its versions that were used.

```
resolveValueSetDefinition(URI valueSetDefinitionURI, String valueSetDefinitionRevisionId,
AbsoluteCodingSchemeVersionReferenceList csVersionList, String versionTag, SortOptionList sortOptionList)
```

Description:	Resolves a value set definition that is available in the system using the supplied set of coding scheme versions.
Input:	<p><i>java.net.URI</i> valueSetDefinitionURI - (Mandatory) value set definition uri to resolve <i>java.lang.String</i> valueSetDefinitionRevisionId - (Optional) version of value set definition <i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReferenceList</i> csVersionList - (Optional) list of coding scheme versions to use in resolution. IF the value set definition uses a version that isn't mentioned in this list, the resolve function will return the codingScheme and version that was used as a default for the resolution. <i>java.lang.String</i> versionTag - (Optional) the tag (e.g. "devel", "production", ...) to be used to determine which coding scheme to be used <i>_org.LexGrid.LexBIG.DataModel.Collections.SortOptionList</i> sortOptionList - (Optional) List of sort options to apply during resolution. If supplied, the sort algorithms will be applied in the order provided. Any algorithms not valid to be applied in context of node set iteration, as specified in the sort extension description, will result in a parameter exception. Available algorithms can be retrieved through the LexBIGService <code>getSortExtensions()</code> method after being defined to the LexBIGServiceManager extension registry.</p>
Output:	<i>org.lexgrid.valuesets.dto.ResolvedValueSetDefinition</i> - Resolved Value Set Definition
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:	<p>Implementation: Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to resolve supplied value set definition URI that is loaded in the system 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 Set Definition. Returns object ResolvedValueSetDefinition, which contains AbsoluteCodingSchemeVersionReferenceList, which tells the Coding Scheme and the versions that were used for this resolve, plus the ResolvedConceptReferencesIterator, which is an iterator for all the valid concepts that belong to the Value Set and Value Set Definition itself.</p> <p>Sample Call:</p> <ul style="list-style-type: none"> Step 1: Instantiate LexEVSVValueSetDefinitionServices if it is not done yet : <pre>grid.valuesets.impl.LexEVSVValueSetDefinitionServices vsdServ = org.lexgrid.valuesets.impl.LexEVSVValueSetDefinitionServicesImpl.defaultInstance();</pre> <ul style="list-style-type: none"> Step 2: Create AbsoluteCodingSchemeVersionReferenceList: <pre>AbsoluteCodingSchemeVersionReferenceList csvList = new AbsoluteCodingSchemeVersionReferenceList(); csvList.addAbsoluteCodingSchemeVersionReference(Constructors.createAbsoluteCodingSchemeVersionReference("Automobiles", "2.0")); csvList.addAbsoluteCodingSchemeVersionReference(Constructors.createAbsoluteCodingSchemeVersionReference("AutomobilesParts", "2.0"));</pre> <ul style="list-style-type: none"> Step 3: Call resolveValueSetDefinition method: <pre>ResolvedValueSetDefinition rvdDef = vsdServ.resolveValueSetDefinition(new URI("SRITEST:AUTO:AllDomesticANDGM1"), null, csvList, "PRODUCTION", null);</pre>
--------------------------------	--

Resolving Supplied Value Set Definition Objects

This function resolves the value set definition that is supplied against the supplied coding scheme version(s) and returns set of valied concept codes plus the code systems and its versions that were used.

```
resolveValueSetDefinition(ValueSetDefinition vsDef, AbsoluteCodingSchemeVersionReferenceList csVersionList,
String versionTag, SortOptionList sortOptionList)
```

Description:	Resolves a value set definition that is supplied using the supplied set of coding scheme version(s).
Input:	<p><i>org.LexGrid.valueSets.ValueSetDefinition</i> vsDef- (Mandatory) value set definition object to resolve</p> <p><i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReferenceList</i> csVersionList - (Optional) list of coding scheme versions to use in resolution. IF the value set definition uses a version that isn't mentioned in this list, the resolve function will return the codingScheme and version that was used as a default for the resolution.</p> <p><i>java.lang.String</i> versionTag - (Optional) the tag (e.g. "devel", "production", ...) to be used to determine which coding scheme to be used</p> <p><i>_org.LexGrid.LexBIG.DataModel.Collections.SortOptionList</i> sortOptionList - (Optional) List of sort options to apply during resolution. If supplied, the sort algorithms will be applied in the order provided. Any algorithms not valid to be applied in context of node set iteration, as specified in the sort extension description, will result in a parameter exception. Available algorithms can be retrieved through the LexBIGService getSortExtensions() method after being defined to the LexBIGServiceManager extension registry.</p>
Output:	<i>org.lexgrid.valuesets.dto.ResolvedValueSetDefinition</i> - Resolved Value Set Definition
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:

Implementation:

Step 1: Call this method on the associated LexEVS Value Set Definition Service instance to resolve supplied value set definition object 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 Set Definition. Returns object ResolvedValueSetDefinition, which contains AbsoluteCodingSchemeVersionReferenceList, which tells the Coding Scheme and the versions that were used for this resolve, plus the ResolvedConceptReferencesIterator, which is an iterator for all the valid concepts that belong to the Value Set and Value Set Definition itself.

Sample Call:

- **Step 1:** Instantiate LexEVSVValueSetDefinitionServices if it is not done yet :

```
grid.valuesets.impl.LexEVSVValueSetDefinitionServices vsdServ = org.lexgrid.valuesets.  
impl.LexEVSVValueSetDefinitionServicesImpl.defaultInstance();
```

- **Step 2:** Generate ValueSetDefinition object:

```
org.LexGrid.valuesets.ValueSetDefinition vsd = new org.LexGrid.valuesets.  
ValueSetDefinition();  
vsd.setValueSetDefinitionURI("SRITEST:JUNIT:TEST:VSD1");  
vsd.setValueSetDefinitionName("JUnit test vsd 1");  
vsd.setConceptDomain("cd");  
vsd.addRepresentsRealmOrContext("context");  
vsd.setDefaultCodingScheme("Automobiles");  
vsd.setIsActive(false);  
vsd.setOwner("lexevs");  
java.util.List<org.LexGrid.commonTypes.Source> srcList = new java.util.ArrayList<org.  
LexGrid.commonTypes.Source>();  
org.LexGrid.commonTypes.Source src = new org.LexGrid.commonTypes.Source();  
src.setContent("lexevs");  
srcList.add(src);  
src = new org.LexGrid.commonTypes.Source();  
src.setContent("cts2");  
srcList.add(src);  
vsd.setSource(srcList);  
  
org.LexGrid.valuesets.DefinitionEntry de = new org.LexGrid.valuesets.DefinitionEntry();  
de.setRuleOrder(1L);  
de.setOperator(org.LexGrid.valuesets.types.DefinitionOperator.OR);  
org.LexGrid.valuesets.CodingSchemeReference csr = new org.LexGrid.valuesets.  
CodingSchemeReference();  
csr.setCodingScheme("Automobiles");  
de.setCodingSchemeReference(csr);  
  
vsd.addDefinitionEntry(de);
```

- **Step 3:** Create AbsoluteCodingSchemeVersionReferenceList:

```
AbsoluteCodingSchemeVersionReferenceList csvList = new  
AbsoluteCodingSchemeVersionReferenceList();  
csvList.addAbsoluteCodingSchemeVersionReference(Constructors.  
createAbsoluteCodingSchemeVersionReference("Automobiles", "2.0"));  
csvList.addAbsoluteCodingSchemeVersionReference(Constructors.  
createAbsoluteCodingSchemeVersionReference("AutomobilesParts", "2.0"));
```

- **Step 4:** Call resolveValueSetDefinition method:

```
ResolvedValueSetDefinition rvdDef = vsdServ.resolveValueSetDefinition(vsd, csvList,  
"PRODUCTION", null);
```

Resolved Value Set Objects

ResolvedValueSetCodedNodeSet

Contains Coding Scheme Version reference list that was used to resolve the Value Set Definition and the CodedNodeSet.

**Note**

The CodedNodeSet is not resolved.

ResolvedValueSetDefinition

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

Error Handling

LexEVS Value Set Definition 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 the console.

Load Scripts

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

- `LoadValueSetDefinition.bat` for Windows environment and
- `LoadValueSetDefinition.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 LoadValueSetDefinition.sh -in "file:///path/to/file.xml"
```

Sample Value Set Definition XML File

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

```

<source>
<lgVD:valueSetDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://LexGrid.org/schema/2010/01/LexGrid/versions http://LexGrid.org/schema/2010/01/LexGrid/versions.xsd"
    xmlns="http://LexGrid.org/schema/2010/01/LexGrid/versions"
    xmlns:lgVer="http://LexGrid.org/schema/2010/01/LexGrid/versions"
    xmlns:lgCommon="http://LexGrid.org/schema/2010/01/LexGrid/commonTypes"
    xmlns:data="data"
    xmlns:lgVD="http://LexGrid.org/schema/2010/01/LexGrid/valueSets"
    xmlns:lgNaming="http://LexGrid.org/schema/2010/01/LexGrid/naming"
    valueSetDefinitionURI="SRITEST:AUTO:PropertyRefTest1-VSDONLY" valueSetDefinitionName="Property Reference
Test 1" defaultCodingScheme="Automobiles" effectiveDate="2009-01-01T11:00:00Z" isActive="true" status="ACTIVE">
    <lgVD:mappings>
        <lgNaming:supportedCodingScheme localId="Automobiles" uri="urn:oid:1.1.1.0.1"></lgNaming:supportedCodingScheme>
        <lgNaming:supportedConceptDomain localId="Autos">Autos</lgNaming:supportedConceptDomain>
        <lgNaming:supportedSource localId="GM">GM</lgNaming:supportedSource>
        <lgNaming:supportedSource localId="Ford">Ford</lgNaming:supportedSource>
    </lgVD:mappings>
    <lgVD:source role="role1" subRef="subRef1">GM</lgVD:source>
    <lgVD:source role="role2" subRef="subRef2">Ford</lgVD:source>
    <lgVD:representsRealmOrContext>Automobiles</lgVD:representsRealmOrContext>
    <lgVD:representsRealmOrContext>Ford</lgVD:representsRealmOrContext>
    <lgVD:representsRealmOrContext>GM</lgVD:representsRealmOrContext>
    <lgVD:properties>
        <lgCommon:property propertyName="textualPresentation">
            <lgCommon:source role="PropRole1" subRef="PropSubRef1">GM</lgCommon:source>
            <lgCommon:source role="PropRole2" subRef="PropSubRef2">Ford</lgCommon:source>
            <lgCommon:usageContext>PropUsageContext GM</lgCommon:usageContext>
            <lgCommon:usageContext>PropUsageContext Ford</lgCommon:usageContext>
            <lgCommon:propertyQualifier propertyQualifierName="PropQual1Namuuuuu" propertyQualifierType="PQual
Type 1">
                <lgCommon:value dataType="text">PropQualValue GM</lgCommon:value>
            </lgCommon:propertyQualifier>
            <lgCommon:propertyQualifier propertyQualifierName="PropQual2Namuuuuu" propertyQualifierType="PQual
Type 2">
                <lgCommon:value dataType="text">PropQualValue Ford</lgCommon:value>
            </lgCommon:propertyQualifier>
            <lgCommon:value>All Automobiles nodes</lgCommon:value>
        </lgCommon:property>
    </lgVD:properties>
    <lgVD:definitionEntry ruleOrder="1" operator="OR">
        <lgVD:propertyReference codingScheme="Automobiles">
            <lgVD:propertyMatchValue matchAlgorithm="LuceneQuery">General</lgVD:propertyMatchValue>
        </lgVD:propertyReference>
    </lgVD:definitionEntry>

</lgVD:valueSetDefinition>
</source>

```

System Testing

The System test case for the LexEVS Value Set Definition 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 Set Definition Services are integrated parts of core LexEVS API and are packaged and installed with other LexEVS services.

Value Set GUI

Visit [LexEVS 6.0 Value Set GUI](#)  for detailed functionality and how tos about using Value Sets developer GUI tool