

4 - LexEVS 5.x Pick List Service

Contents of this Page

- [Introduction](#)
- [Pick List Class Diagram](#)
- [LexEVS Pick List Service API](#)
 - [Administrative Functions](#)
 - [Loading Pick List Definitions](#)
 - [loadPickList\(PickListDefinition pldef, String systemReleaseURI\)](#)
 - [loadPickList\(InputStream inputStream, boolean failOnAllErrors\)](#)
 - [loadPickList\(String xmlFileLocation, boolean failOnAllErrors\)](#)
 - [Remove Pick List Definition](#)
 - [Validate XML Resources](#)
 - [Query Functions](#)
 - [getPickListDefinitionById](#)
 - [getPickListDefinitionsForDomain](#)
 - [getPickListValueDomain](#)
 - [listPickListIds](#)
 - [resolvePickList](#)
 - [resolvePickListForTerm](#)
 - [Resolved Pick List Objects](#)
 - [ResolvedPickListEntry](#)
 - [ResolvedPickListEntryList](#)
 - [Error Handling](#)
- [Load Scripts](#)
- [Sample Pick List Definitions XML File](#)
- [Installation / Packaging](#)
- [System Testing](#)

Introduction

This document is a section of the [LexEVS 5.x Programmer's Guide](#). It is new for LexEVS v5.1.

The Pick List Services are integrated parts of the LexEVS core API. They provide the ability to:

- Load Pick List definitions programmatically into the LexGrid repository using the domain objects that are available via the LexGrid logical model
- Apply user restrictions (ex: pickListId) and dynamically resolve the definitions at run time

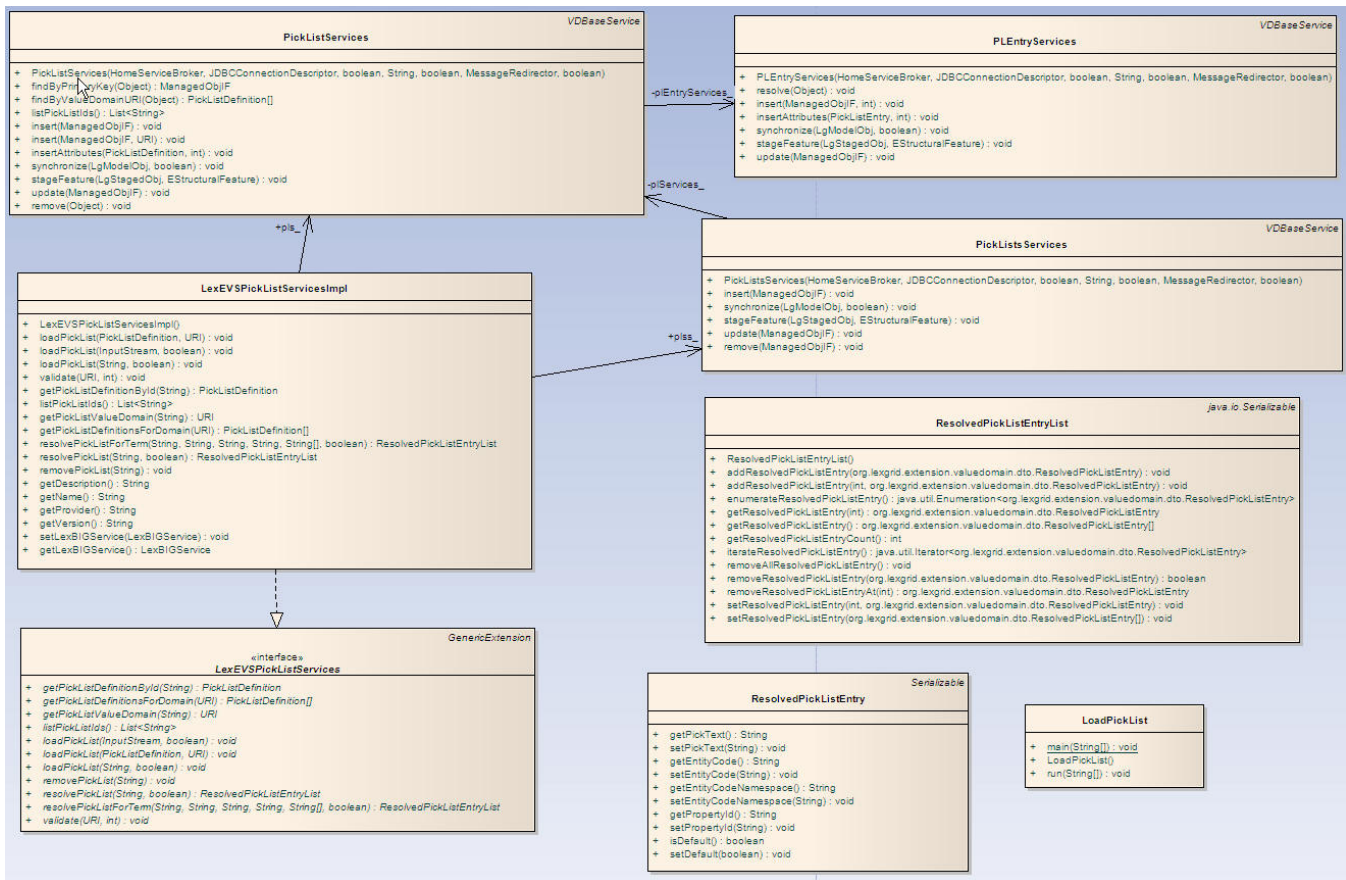
The LexEVS Pick List Services expose the API particularly for the Pick List elements of the LexGrid Logical Model. For more information on LexGrid model see <http://informatics.mayo.edu/>.

Pick List Class Diagram

These classes implement the LexEVS Pick List API.

Class Name	Description
PickListsServices	Class to handle list of Pick List Definitions.
PickListServices	Class to handle individual Pick List Definition objects to and fro database.
PEntryServices	Class to handle Pick List Entry objects to and fro database.
LexEVSPickListServices	Primary interface for LexEVS Pick List API.
LexEVSPickListServicesImpl	Implementation of LexEVSPickListServices which is primary interface for LexEVS Pick List API.
LoadPickList	Imports the Pick List Definitions in the source file, provided in LexGrid canonical format, to the LexBIG repository.
ResolvedPickListEntryList	Class to hold list of resolved pick list entries.
ResolvedPickListEntry	Bean for resolved pick list entries.

The following figure shows the classes in a diagram.



LexEVS Pick List Service API

Administrative Functions

LexEVS PickListServices provides administrative functions to load and remove pick list definitions.

Loading Pick List Definitions

There are three methods that could be used to load Pick List Definitions:

- loadPickList(PickListDefinition pldef, String systemReleaseURI)
- loadPickList(InputStream inputStream, boolean failOnAllErrors)
- loadPickList(String xmlFileLocation, boolean failOnAllErrors)

loadPickList(PickListDefinition pldef, String systemReleaseURI)

Description:	Loads supplied Pick List Definition object
Input:	org.LexGrid.emf.valueDomains.PickListDefinition, String
Output:	none
Exception:	LBException

Implementation Details:	Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Pick List Service instance to load the Pick List Definition object and the System Release URI that this definition belongs to. Sample Call: • <i>Step 1:</i> Instantiate LexEVSPickListServices if it is not done yet : <pre>org.lexgrid.valuedomain.LexEVSPickListServices pls = new org.lexgrid.valuedomain.impl.LexEVSPickListServicesImpl();</pre> • <i>Step 2:</i> Create and populate the PickListDefinition object. <pre>PickListDefinition can be created using _org.LexGrid.emf.valueDomains.PickListDefinition pickList = org.LexGrid.emf.valueDomains.ValuedomainsFactory.eINSTANCE.createPickListDefinition(); Data for pickList object can be populated by using set methods: pickList.setPickListId(pickListId); pickList.setRepresentsValueDomain(vdURI); pickList.setCompleteDomain(true); pickList.setDefaultEntityCodeNamespace(ecns); pickList.setDefaultLanguage("en"); pickList.setDefaultSortOrder("asc"); pickList.setIsActive(true); pickList.setEntityDescription(red);</pre> Similarly, PickListEntryNode, Property, and Mapping objects can be created and assigned to the pickList object. <pre>pickList.getPickListEntryNode.add(pickListEntry); pickList.setProperties(propertisObject); pickList.setMappings(mappingsObject);</pre> • <i>Step 3:</i> Call the load method by passing the Pick List Definition object and the System Release URI. <pre>pls.loadPickList(pickList, "Release 2009");</pre>
--------------------------------	--

loadPickList(InputStream inputStream, boolean failOnAllErrors)

Description:	Loads Pick List Definitions found in inputStream.
Input:	<i>java.io.InputStream</i> <i>boolean</i>
Output:	none
Exception:	<i>Exception</i>
Implementation Details:	Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Pick List Service instance to load all Pick List Definitions from the inputstream. Sample Call: • <i>Step 1:</i> Instantiate LexEVSPickListServices if it is not done yet: <pre>org.lexgrid.valuedomain.LexEVSPickListServices pls = new org.lexgrid.valuedomain.impl.LexEVSPickListServicesImpl();</pre> • <i>Step 2:</i> Call load method by passing the inputStream and boolean flag whether to stop on load errors: <i>pls.loadPickList(inputStream, true);</i>

loadPickList(String xmlFileLocation, boolean failOnAllErrors)

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

Remove Pick List Definition

removePickList(String pickListId)

Description:	Removes supplied Pick List Definition from the system.
Input:	<i>java.lang.String</i>
Output:	none
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException,</i> <i>org.LexGrid.managedobj.RemoveException</i>
Implementation Details:	Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Pick List Service instance to remove Pick List Definition from the system that matches the supplied pickListId. Sample Call: <i>Step 1:</i> Instantiate LexEVSPickListServices if it is not done yet : <pre>org.lexgrid.valuedomain.LexEVSPickListServices pls = new org.lexgrid.valuedomain.impl.LexEVSPickListServicesImpl();</pre> <i>Step 2:</i> Call removePickList method: <pre>vds.removePickList ("AUTO:AllDomesticANDGM");</pre>

Validate XML Resources

validate

Description:	Perform validation of the candidate resource without loading data.
Input:	<i>java.net.URI</i> <i>int</i>
Output:	none
Exception:	<i>Org.LexGrid.LexBIG.Exceptions.LBParameterException</i>

Implementation Details:	Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Pick List Service instance to validate the XML file that is in LexGrid format. This call will not load the data in an XML file. Sample Call: <i>Step 1:</i> Instantiate LexEVSPickListServices if it is not done yet : <pre>org.lexgrid.valuedomain.LexEVSPickListServices pls = new org.lexgrid.valuedomain.impl.LexEVSPickListServicesImpl();</pre> <i>Step 2:</i> Call validate method for validation by supplying URI of the XML file and validation level. Supported validationLevels include: <pre>0 = Verify document is well-formed 1 = Verify document is valid _pls.validate(uriOfXMLFile, true);</pre>
--------------------------------	---

Query Functions

Following are some of the query functions provided by LexEVS PickListServices.

getPickListDefinitionById

getPickListDefinitionById(String pickListId)

Description:	Returns pickList definition for supplied pickListId.
Input:	<i>java.lang.String</i>
Output:	<i>org.LexGrid.emf.valueDomains.PickListDefinition</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Pick List Service instance to get Pick List Definition for supplied pickListId. Sample Call: <i>Step 1:</i> Instantiate LexEVSPickListServices if it is not done yet : <pre>org.lexgrid.valuedomain.LexEVSPickListServices pls = new org.lexgrid.valuedomain.impl.LexEVSPickListServicesImpl();</pre> <i>Step 2:</i> Call getPickListDefinitionById method: <pre>PickListDefinition plDef = pls.getPickListDefinitionById("AUTO:DomesticAutoMakers");</pre>

getPickListDefinitionsForDomain

getPickListDefinitionsForDomain(URI valueDomainURI)

Description:	Returns all the pickList definitions that represent the supplied valueDomain URI.
Input:	<i>java.net.URI</i>
Output:	<i>org.LexGrid.emf.valueDomains.PickListDefinition[]</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:	Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Pick List Service instance to get all the Pick List Definitions that are represented by supplied Value Domain URI. Sample Call: <i>Step 1:</i> Instantiate LexEVSPickListServices if it is not done yet : <pre>org.lexgrid.valuedomain.LexEVSPickListServices pls = new org.lexgrid.valuedomain.impl.LexEVSPickListServicesImpl();</pre> <i>Step 2:</i> Call getPickListDefinitionsForDomain method: <pre>PickListDefinition[] plDefs = pls.getPickListDefinitionsForDomain(valueDomainURI);</pre>
--------------------------------	--

getPickListValueDomain

getPickListValueDomain(String pickListId)

Description:	Returns a URI of the represented valueDomain of the pickList.
Input:	<i>java.lang.String</i>
Output:	<i>java.net.URI</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Pick List Service instance to get a Value Domain URI represented by supplied pickListId. Sample Call: <i>Step 1:</i> Instantiate LexEVSPickListServices if it is not done yet : <pre>org.lexgrid.valuedomain.LexEVSPickListServices pls = new org.lexgrid.valuedomain.impl.LexEVSPickListServicesImpl();</pre> <i>Step 2:</i> Call getPickListValueDomain method: <pre>URI vdURI = pls.getPickListValueDomain ("AUTO:DomesticAutoMakers");</pre>

listPickListIds

listPickListIds()

Description:	Returns a list of pickListIds that are available in the system.
Input:	none
Output:	<i>java.util.List<java.lang.String></i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:	Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Pick List Service instance to get all the PickListIds that are loaded in the system. Sample Call: <i>Step 1:</i> Instantiate LexEVSPickListServices if it is not done yet : <pre>org.lexgrid.valuedomain.LexEVSPickListServices pls = new org.lexgrid.valuedomain.impl.LexEVSPickListServicesImpl();</pre> <i>Step 2:</i> Call listPickListIds method: <pre>List<String> pList = pls.listPickListIds();</pre>
--------------------------------	---

resolvePickList

resolvePickList(String pickListId, boolean sortByText)

Description:	Resolves pickList definition for supplied pickListId.
Input:	<i>java.lang.String, boolean</i>
Output:	<i>org.lexgrid.valuedomain.dto.ResolvedPickListEntryList</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>
Implementation Details:	Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Pick List Service instance to get the resolved Pick List Entries for the supplied pickListId. Optionally, if sortByTests is true, sort the pickText in the list. Sample Call: <i>Step 1:</i> Instantiate LexEVSPickListServices if it is not done yet : <pre>org.lexgrid.valuedomain.LexEVSPickListServices pls = new org.lexgrid.valuedomain.impl.LexEVSPickListServicesImpl();</pre> <i>Step 2:</i> Call resolvePickList method: <pre>ResolvedPickListEntryList pleList = pls.resolvePickList ("AUTO:DomesticAutoMakers", true);</pre>

resolvePickListForTerm

resolvePickListForTerm(String pickListId, String term, String matchAlgorithm, String language, String[] context, boolean sortByText)

Description:	Resolves pickList definition by applying supplied arguments.
Input:	<i>java.lang.String java.lang.String java.lang.String java.lang.String java.lang.String[] boolean</i>
Output:	<i>org.lexgrid.valuedomain.dto.ResolvedPickListEntryList</i>
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Implementation Details:	Implementation: <i>Step 1:</i> Call this method on the associated LexEVS Pick List Service instance to get list of Pick List Entries that match the term supplied and meet other supplied restrictions. Sample Call: • <i>Step 1:</i> Instantiate LexEVSPickListServices if it is not done yet : <pre>org.lexgrid.valuedomain.LexEVSPickListServices pls = new org.lexgrid.valuedomain.impl.LexEVSPickListServicesImpl();</pre> • <i>Step 2:</i> Call resolvePickListForTerm method: <pre>ResolvedPickListEntryList pleList = pls.resolvePickListForTerm ("AUTO:DomesticAutoMakers", "Jaguar", MatchAlgorithms.exactMatch.name(), "en", null, true);</pre>
--------------------------------	---

Resolved Pick List Objects

ResolvedPickListEntry

ResolvedPickListEntry contains resolved Pick List Entry Nodes.

ResolvedPickListEntryList

ResolvedPickListEntryList contains the list of resolved Pick List Entries. Also provides helpful features to add, remove, enumerate Pick List Entries.

Error Handling

Both LexEVS Pick List 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

Scripts to load Pick List Definitions into LexEVS system will be located under 'Admin' folder of LexEVS install directory. These loader scripts will only load data in an XML file that is in LexGrid format.

LoadPickList.bat for Windows environment and LoadPickList.sh for Unix environment.

Both of these 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 LoadPickList.sh \-in "file:///path/to/file.xml"
```

Sample Pick List Definitions XML File

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

<source>

```

<?xml version="1.0" encoding="UTF-8"?>
<systemRelease xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
{panel}
  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>
<pickLists>
  <lgVD:mappings>
    <lgNaming:supportedCodingScheme localId="Automobiles" uri="urn:oid:11.11.0.1">Automobiles</lgNaming:
supportedCodingScheme>
    <lgNaming:supportedLanguage localId="en" uri="www.en.org/orsomething">en</lgNaming:
supportedLanguage>
    <lgNaming:supportedNamespace localId="Automobiles" uri="urn:oid:11.11.0.1" 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:pickListDefinition pickListId="SRITEST:AUTO:DomesticAutoMakers" representsValueDomain="SRITEST:
AUTO:DomesticAutoMakers" isActive="true" defaultEntityCodeNamespace="Automobiles" defaultLanguage="en"
completedDomain="false">
    <lgCommon:owner>Owner for Domestic Auto Makers</lgCommon:owner>
    <lgCommon:entityDescription>DomesticAutoMakers</lgCommon:entityDescription>
    <lgVD:mappings>
      <lgNaming:supportedCodingScheme localId="Automobiles" uri="urn:oid:11.11.0.1">Automobiles<
/ lgNaming:supportedCodingScheme>
      <lgNaming:supportedDataType localId="texthtml">text/html</lgNaming:supportedDataType>
      <lgNaming:supportedDataType localId="textplain">text/plain</lgNaming:supportedDataType>
      <lgNaming:supportedLanguage localId="en" uri="www.en.org/orsomething">en</lgNaming:
supportedLanguage>
      <lgNaming:supportedNamespace localId="Automobiles" uri="urn:oid:11.11.0.1"
equivalentCodingScheme="Automobiles">Automobiles</lgNaming:supportedNamespace>
      <lgNaming:supportedProperty localId="textualPresentation">textualPresentation</lgNaming:
supportedProperty>
      <lgNaming:supportedSource assemblyRule="rule1" uri="http://informatics.mayo.edu" localId="
lexgrid.org">lexgrid.org</lgNaming:supportedSource>
      <lgNaming:supportedSource localId="_111101">11.11.0.1</lgNaming:supportedSource>
    </lgVD:mappings>
    <lgVD:pickListEntryNode pickListEntryId="PLGmp1" isActive="true">
      <lgCommon:owner>Owner for PLGmp1</lgCommon:owner>
      <lgCommon:entryState containingRevision="R001" relativeOrder="1" changeType="NEW" prevRevision="
R00A"/>
      <lgVD:inclusionEntry entityCode="GM" entityCodeNamespace="Automobiles" propertyId="p1">
        <lgVD:pickText>General Motors</lgVD:pickText>
      </lgVD:inclusionEntry>
      <lgVD:properties>
        <lgCommon:property propertyName="textualPresentation" isActive="true" language="en"
propertyId="p1" propertyType="presentation" status="active" effectiveDate="2001-12-17T09:30:47Z"
expirationDate="2011-12-17T09:30:47Z">
          <lgCommon:owner role="role" subRef="subref">General Motors</lgCommon:owner>
          <lgCommon:entryState containingRevision="R001" relativeOrder="1" changeType="NEW"
prevRevision="R00A"/>
          <lgCommon:source subRef="subref1" role="role1">General Motors</lgCommon:source>
          <lgCommon:value dataType="textplain">Property for General Motors</lgCommon:value>
        </lgCommon:property>
      </lgVD:properties>
    </lgVD:pickListEntryNode>
    <lgVD:pickListEntryNode pickListEntryId="PLGmp2" isActive="true">
      <lgCommon:owner>Owner for PLGmp2</lgCommon:owner>
      <lgCommon:entryState containingRevision="R001" relativeOrder="1" changeType="NEW" prevRevision="
R00A"/>
      <lgVD:inclusionEntry entityCode="GM" entityCodeNamespace="Automobiles" propertyId="p2">

```

```

        <lgVD:pickText>GM</lgVD:pickText>
    </lgVD:inclusionEntry>
</lgVD:pickListEntryNode>
<lgVD:pickListEntryNode pickListEntryId="PLJaguarp1" isActive="true">
    <lgCommon:owner>Owner for PLJaguarp1</lgCommon:owner>
    <lgCommon:entryState containingRevision="R001" relativeOrder="1" changeType="NEW" prevRevision="
R00A"/>
    <lgVD:inclusionEntry entityCode="Jaguar" entityCodeNamespace="Automobiles" propertyId="p1">
        <lgVD:pickText>Jaguar</lgVD:pickText>
    </lgVD:inclusionEntry>
</lgVD:pickListEntryNode>
<lgVD:pickListEntryNode pickListEntryId="PLChevroletp1" isActive="true">
    <lgCommon:owner>Owner for PLChevroletp1</lgCommon:owner>
    <lgCommon:entryState containingRevision="R001" relativeOrder="1" changeType="NEW" prevRevision="
R00A"/>
    <lgVD:inclusionEntry entityCode="Chevy" entityCodeNamespace="Automobiles" propertyId="p1">
        <lgVD:pickText>Chevrolet</lgVD:pickText>
    </lgVD:inclusionEntry>
</lgVD:pickListEntryNode>
</lgVD:pickListDefinition>
<lgVD:pickListDefinition pickListId="SRITEST:AUTO:DomasticLeafOnly" representsValueDomain="SRITEST:AUTO:
DomasticLeafOnly" completeDomain="true" defaultEntityCodeNamespace="Automobiles" defaultLanguage="en" isActive="
true">
    <lgCommon:entityDescription>Leaf Only Nodes of Domastic AutoMakers</lgCommon:entityDescription>
</lgVD:pickListDefinition>
</pickLists>
</systemRelease>

```

```
</source>
```

Installation / Packaging

Pick List service are integrated parts of core LexEVS API and are packaged and installed with other LexEVS services.

System Testing

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

```
org.LexGrid.LexBIG.Impl.testUtility.PickListAllTests
```

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