

LexEVS 6.0 CTS2 Authoring 5 - Value Set Authoring Operation API

Contents of this Page

- [Introduction](#)
- [Interface](#)
- [Revision Information](#)
- [Authoring Functions](#)
 - [Create Operations](#)
 - [createValueSet](#)
 - [createValueSet using ValueSetDefinition object](#)
 - [addDefinitionEntry](#)
 - [addValueSetProperty](#)
 - [Edit Operations](#)
 - [updateValueSetMetaData](#)
 - [updateDefinitionEntry](#)
 - [updateValueSetProperty](#)
 - [Remove Operations](#)
 - [removeValueSet](#)
 - [removeDefinitionEntry](#)
 - [removeValueSetProperty](#)
 - [Versionable Change Operations](#)
 - [updateValueSetStatus](#)
 - [updateValueSetVersionable](#)

CTS2 Links for LexEVS 6.0

- [CTS2 API Main Page](#)
- [Programmer's Guide Main Page](#)
- [LexEVS 6.0 Main Page](#)
- [LexEVS Current Release](#)

Introduction

LexEVS CTS2 Value Set Authoring API provides capability to author Value Set Definition, its Properties, Definition Entries (rule set) and its Versionable attributes.

Here are the authoring functions that can be performed on Value Set:

- **Create** - This function provides capability to create:
 - New Value Set
 - Add new Definition Entry (Rule Set) to Value Set
 - Add new property to Value Set
- **Edit** - This function provides capability to update:
 - Value Set meta-data
 - Definition Entry of a Value Set
 - Property of a Value Set
- **Remove** - This function provides capability to remove:
 - Value Set
 - Definition Entry of a Value Set
 - Property of a Value Set
- **Versionable Change** - This function provides capability to modify versionable attributes of Usage Context like Status, Effective Date, Expiration Date, isActive and Owner:
 - Value Set Status
 - Value Set Versionable attributes

Interface

org.lexevs.cts2.author.ValueSetAuthoringOperation is the main interface for all the authoring operations against Value Set. This interface can be accessed using main LexEVSCTS2 interface, like:

```
org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthOp = new org.lexevs.cts2.LexEvsCTS2Impl().  
getAuthoringOperation().getValueSetAuthoringOperation();
```

Revision Information

All the authoring functions described here requires information about the author and revision/version id to be assigned to entities for each of these operations. This is done by passing object **org.lexevs.cts2.core.update.RevisionInfo**. RevisionInfo object has following attributes:

- **java.lang.String changeAgent** - (Optional) The source that participated in this particular change.
- **java.lang.String changeInstruction** - (Optional) A human or machine readable set of instructions on how to apply this change.
- **java.lang.String revisionId** - (**Mandatory**) The unique identifier of this revision.
- **java.lang.Long editOrder** - (Optional) The relative order that this revision is to be applied if in a systemRelease.
- **java.util.Date revisionDate** - (Optional) The end date for which this version is operative (considered committed).
- **java.lang.String description** - (Optional) The description of the resource/change.
- **java.lang.String systemReleaseURI** - (Optional) The official URI of this release

Authoring Functions

Following sections contains detailed functions provided by ValueSetAuthoringOperation interface.

Create Operations

Create Operation provides capability to create Value Set, Definition Entry and Properties. For every entry that gets created, a unique revision(version) identifier will be assigned to that entry. This helps in retrieving snapshots of an entry based on its revision(version) identifier. This unique id can be passed in using the RevisionInfo object described above.

createValueSet

This function creates a new Value Set. The new Value Set is defined by the set of meta-data properties that describe it.

```
createValueSet(URI valueSetURI, String valueSetName, String defaultCodeSystem, String conceptDomainId, List<org.LexGrid.commonTypes.Source> sourceList, List<String> usageContextList, Properties properties, List<DefinitionEntry> ruleSetList, Versionable versionable, RevisionInfo revision)
```

Description:	Creates a new value set definition and loads into repository.
Input:	• java.net.URI valueSetURI - (Mandatory) URI of value set definition. • java.lang.String valueSetName - (Mandatory) Name of value set definition. • java.lang.String defaultCodeSystem - (Optional) local name of default code system. • java.lang.String conceptDomainId - (Optional) local name of concept domain this value set is binded to. • java.util.List<org.LexGrid.commonTypes.Source> sourceList - (Optional)Source list for this value set. • java.util.List<java.lang.String> usageContextList - (Optional) list of local name of usage context of this value set. • org.LexGrid.commonTypes.Properties properties - (Optional) list of value set properties. • java.util.List<org.LexGrid.valuesets.DefinitionEntry> ruleSetList - (Optional) list of definition entries (rule sets). • org.LexGrid.commonTypes.Versionable versionable - (Optional) Versionable entries of a value set(status, isActive, effective date etc). • org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	java.net.URI - value set definition URI if created successfully
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	Step 1: Instantiate ValueSetAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();</pre>
	Step 2: Populate RevisionInfo object: <pre>RevisionInfo revInfo = new RevisionInfo(); revInfo.setChangeAgent("change Agent Name"); revInfo.setChangeInstruction("here goes the change Instructions"); revInfo.setDescription("description of the resource"); revInfo.setEditOrder(1L); revInfo.setRevisionDate(new Date()); revInfo.setRevisionId(UUID.randomUUID().toString());</pre>
	Step 3: Populate a properties to be added: <pre>Property prop = new Property(); prop.setPropertyId("propertyId1"); prop.setPropertyName("propertyName"); prop.setIsActive(false); prop.setLanguage("en"); prop.setOwner("owner"); prop.setPropertyType(PropertyTypes.PROPERTY.name()); Text text = new Text(); text.setContent("content"); text.setDataTypes("Text datatype"); prop.setValue(text); Properties props = new Properties(); props.addProperty(prop);</pre>
	Step 4: call create Value Set method: <pre>'URI vsdURI = vsAuthOp.createValueSet(new URI("VSD:AUTHORING:JUNIT:TEST1"), "Authoring create vsd junit test1", "Automobiles", "Autos", null, null, props, null, null, revInfo);'</pre>

createValueSet using ValueSetDefinition object

This function is similar to above except this allows users to create Value Set by passing well defined value set definition object. The new Value Set is defined by the set of meta-data properties that describe it.

createValueSet(ValueSetDefinition valueSetDefininition, RevisionInfo revision)

Description:	Creates a new value set definition and loads into repository.
Input:	org.LexGrid.valuesets.ValueSetDefinition valueSetDefininition - (Mandatory) Value Set Definition object to be loaded into repository. org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	<i>java.net.URI</i> - value set definition URI if created successfully
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	Step 1: Instantiate ValueSetAuthoringOperation if it is not done yet:
	<pre>org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();</pre>
	Step 2: Populate RevisionInfo object:
	<pre>RevisionInfo revInfo = new RevisionInfo(); revInfo.setChangeAgent("change Agent Name"); revInfo.setChangeInstruction("here goes the change Instructions"); revInfo.setDescription("description of the resource"); revInfo.setEditOrder(1L); revInfo.setRevisionDate(new Date()); revInfo.setRevisionId(UUID.randomUUID().toString());</pre>
	Step 3: Populate a Value Set Definition to be added:
	<pre>ValueSetDefinition vsd = new ValueSetDefinition(); vsd.setValueSetDefinitionURI("VSD:AUTHORING:JUNIT:TEST2"); vsd.setValueSetDefinitionName("Authoring create vsd junit test2"); vsd.setConceptDomain("Autos CD"); vsd.setDefaultCodingScheme("Automobiles");</pre>
	Step 4: call create Value Set method:
	<pre>URI vsdURI = vsAuthorOp.createValueSet(vsd, revInfo); </pre>

addDefinitionEntry

This function provides capability to add new Definition Entry (Rule Set) to Value Set Definition.

addDefinitionEntry(URI valueSetURI, DefinitionEntry newDefinitionEntry, RevisionInfo revision)

Description:	Add new definition entry (rule set) to a value set definition.
Input:	- java.net.URI valueSetURI - (Mandatory) URI of value set definition. - org.LexGrid.valuesets.DefinitionEntry newDefinitionEntry - (Mandatory) Definition Entry object to be added. - org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	boolean - True if update was successful
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:

• Step 1: Instantiate ValueSetAuthoringOperation if it is not done yet:

```
org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();
```

• Step 2: Populate RevisionInfo object:

```
RevisionInfo revInfo = new RevisionInfo();
revInfo.setChangeAgent("change Agent Name");
revInfo.setChangeInstruction("here goes the change Instructions");
revInfo.setDescription("description of the resource");
revInfo.setEditOrder(1L);
revInfo.setRevisionDate(new Date());
revInfo.setRevisionId(UUID.randomUUID().toString());
```

• Step 3: Populate a Definition Entry object to be added:

```
DefinitionEntry newDefEntry = new DefinitionEntry();
newDefEntry.setOperator(org.LexGrid.valuesets.types.DefinitionOperator.OR);
newDefEntry.setRuleOrder(0L);

org.LexGrid.valuesets.EntityReference er = new org.LexGrid.valuesets.EntityReference();
er.setEntityCode("GM");
er.setEntityCodeNamespace("Automobiles");
er.setLeafOnly(false);
er.setReferenceAssociation("hasSubtype");
er.setTargetToSource(false);
er.setTransitiveClosure(true);

newDefEntry.setEntityReference(er);
```

• Step 4: call create definition entry method by passing Value Set URI and Definition Entry object:

```
boolean created = vsAuthOp.addDefinitionEntry(new URI("VSD:AUTHORING:JUNIT:TEST2"),
newDefEntry, revInfo);
```

addValueSetProperty

This function provides capability to add a new property to a Value Set.

addValueSetProperty(URI valueSetURI, Property newProperty, RevisionInfo revision)

Description:	Add new Property to existing value set definition.
Input:	<i>java.net.URI valueSetURI</i> - (Mandatory) URI of value set definition this new property will be added. <i>org.LexGrid.commonTypes.Property newProperty</i> - (Mandatory) New property that will be added to Value Set Definition. <i>org.lexevs.cts2.core.update.RevisionInfo revision</i> - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	<i>boolean</i> - True; if addition of new property was success
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	Step 1: Instantiate ValueSetAuthoringOperation if it is not done yet:
	<pre>org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();</pre>
	Step 2: Populate RevisionInfo object:
	<pre>RevisionInfo revInfo = new RevisionInfo(); revInfo.setChangeAgent("change Agent Name"); revInfo.setChangeInstruction("here goes the change Instructions"); revInfo.setDescription("description of the resource"); revInfo.setEditOrder(1L); revInfo.setRevisionDate(new Date()); revInfo.setRevisionId(UUID.randomUUID().toString());</pre>
	Step 3: Populate a property to be added:
	<pre>Property prop = new Property(); prop.setPropertyId("propertyId1"); prop.setPropertyName("propertyName"); prop.setIsActive(false); prop.setLanguage("en"); prop.setOwner("owner"); prop.setPropertyType(PropertyTypes.PROPERTY.name()); Text text = new Text(); text.setContent("content"); text.setDataTypes("Text datatype"); prop.setValue(text);</pre>
	Step 4: call add property method by passing the Value Set information and a new property:
	<pre>'boolean added = vsAuthOp.addValueSetProperty(new URI("VSD:AUTHORING:JUNIT:TEST1"), prop, revInfo);'</pre>

Edit Operations

The edit operation provides the capability to modify Value Set meta-data, it's definition entires (rule sets) and its properties. For every entry that gets modified, a unique revision(version) identifier will be assigned to that entry. This helps in retrieving snapshots of an entry based on its revision(version) identifier. This unique id can be passed in using the RevisionInfo object described above.

updateValueSetMetaData

This function provides capability to modify existing meta-data of a value set definition.

updateValueSetMetaData(URI valueSetURI, String valueSetName, String defaultCodeSystem, String conceptDomainId, List<org.LexGrid.commonTypes.Source> sourceList, List<String> usageContext, RevisionInfo revision)

Description:	Updates value set definition meta data.
Input:	java.net.URI valueSetURI - (Mandatory) URI of value set definition that will be modified. java.lang.String valueSetName - (Optional) modified value set definition name. Pass 'null' for no change for this attribute. java.lang.String defaultCodeSystem - (Optional) modified default code system name. Pass 'null' for no change for this attribute. java.lang.String conceptDomainId - (Optional) modified concept domain id. Pass 'null' for no change for this attribute. java.util.List<org.LexGrid.commonTypes.Source> sourceList - (Optional) modified complete list of source. Pass 'null' for no change for this attribute. java.util.List<java.lang.String> usageContext - (Optional) modified complete list of context. Pass 'null' for no change for this attribute. org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	boolean - True; if update was success
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:

• Step 1: Instantiate ValueSetAuthoringOperation if it is not done yet:

```
org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();
```

• Step 2: Populate RevisionInfo object:

```
RevisionInfo revInfo = new RevisionInfo();
revInfo.setChangeAgent("change Agent Name");
revInfo.setChangeInstruction("here goes the change Instructions");
revInfo.setDescription("description of the resource");
revInfo.setEditOrder(1L);
revInfo.setRevisionDate(new Date());
revInfo.setRevisionId(UUID.randomUUID().toString());
```

• Step 3: Populate a modified meta-data. Here, just modifying source and usage context:

```
List<org.LexGrid.commonTypes.Source> srcList = new ArrayList<org.LexGrid.commonTypes.Source>();
org.LexGrid.commonTypes.Source src = new org.LexGrid.commonTypes.Source();
src.setContent("LexEVS as src");
srcList.add(src);
src = new Source();
src.setContent("Mayo as src");
srcList.add(src);

List<String> contextList = new ArrayList<String>();
contextList.add("Autos as context");
contextList.add("lexevs as context");
```

• Step 4: call update meta-data method by passing the value set information and the modified meta-data:

```
'boolean updated = vsAuthOp.updateValueSetMetaData(new URI("VSD:AUTHORING:JUNIT:TEST2"),
                                                    "VSD after metatdate update vsdR250",
                                                    null, "Autos CD update vsdR250", srcList, contextList,
                                                    revInfo);'
```

updateDefinitionEntry

This function provides capability to modify existing Definition Entry of a Value Set Definition.

updateDefinitionEntry(URI valueSetURI, DefinitionEntry changedDefinitionEntry, RevisionInfo revision)

Description:	Update existing definition entry (rule set) of a value set definition.
Input:	- java.net.URI valueSetURI - (Mandatory) URI of value set definition that contains the definition entry. - org.LexGrid.valueSetDefinitionEntry changedDefinitionEntry - (Mandatory) modified definition entry object. - org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	boolean - True; if update was success
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	Step 1: Instantiate ValueSetAuthoringOperation if it is not done yet:
	<pre>org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();</pre>
	Step 2: Populate RevisionInfo object:
	<pre>RevisionInfo revInfo = new RevisionInfo(); revInfo.setChangeAgent("change Agent Name"); revInfo.setChangeInstruction("here goes the change Instructions"); revInfo.setDescription("description of the resource"); revInfo.setEditOrder(1L); revInfo.setRevisionDate(new Date()); revInfo.setRevisionId(UUID.randomUUID().toString());</pre>
	Step 3: Populate a modified Definition Entry object:
	<pre>DefinitionEntry changedDefinitionEntry = new DefinitionEntry(); changedDefinitionEntry.setRuleOrder(0L); changedDefinitionEntry.setOperator(DefinitionOperator.OR); EntityReference er = new EntityReference(); er.setEntityCode("GM"); er.setEntityCodeNamespace("Automobiles"); // change leafOnly to true and association type to "CHD"_ er.setLeafOnly(true); er.setReferenceAssociation("CHD");</pre>
	Step 4: call update definition entry method by passing the value set information and the modified definition entry object:
	<pre>boolean updated = vsAuthOp.updateDefinitionEntry(new URI("VSD:AUTHORING:JUNIT:TEST2"), changedDefinitionEntry, revInfo);</pre>

updateValueSetProperty

This function provides capability to modify existing property of a Value Set Definition.

updateValueSetProperty(URI valueSetURI, Property changedProperty, RevisionInfo revision)

Description:	Updates existing property of a value set definition.
Input:	java.net.URI valueSetURI - (Mandatory) URI of value set definition that contains the property. org.LexGrid.commonTypes.Property property - (Mandatory) Modified property. org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	boolean - True; if update was success
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	Step 1: Instantiate ValueSetAuthoringOperation if it is not done yet:
	<pre>org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();</pre>
	Step 2: Populate RevisionInfo object:
	<pre>RevisionInfo revInfo = new RevisionInfo(); revInfo.setChangeAgent("change Agent Name"); revInfo.setChangeInstruction("here goes the change Instructions"); revInfo.setDescription("description of the resource"); revInfo.setEditOrder(1L); revInfo.setRevisionDate(new Date()); revInfo.setRevisionId(UUID.randomUUID().toString());</pre>
	Step 3: Populate a modified property. Just modified isActive to 'true' and language to 'eng' in property 'propertyId1':
	<pre>Property prop = new Property(); prop.setPropertyId("propertyId1"); prop.setPropertyName("propertyName"); prop.setIsActive(true); prop.setLanguage("eng");</pre>
	Step 4: call update property method by passing the value set information and the modified property:
	<pre>boolean updated = vsAuthOp.updateValueSetProperty(new URI("VSD:AUTHORING:JUNIT:TEST1"), prop, revInfo);</pre>

Remove Operations

Remove operation provides capability to remove Value Set, its definition entry and its property.

removeValueSet

This function provides capability to remove a Value Set. This operation is permanent removal of value set definition.

removeValueSet(URI valueSetURI, RevisionInfo revision)

Description:	Removes value set definition from the system. This operation is permanent removal of value set definition.
Input:	java.net.URI valueSetURI - (Mandatory) URI of value set definition to be removed. org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	boolean - True; if remove was success
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	Step 1: Instantiate ValueSetAuthoringOperation if it is not done yet:
	<pre>org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();</pre>
	Step 2: Populate RevisionInfo object:
	<pre>RevisionInfo revInfo = new RevisionInfo(); revInfo.setChangeAgent("change Agent Name"); revInfo.setChangeInstruction("here goes the change Instructions"); revInfo.setDescription("description of the resource"); revInfo.setEditOrder(1L); revInfo.setRevisionDate(new Date()); revInfo.setRevisionId(UUID.randomUUID().toString());</pre>
	Step 3: call remove Value Set method by passing the value set information:
	<pre>boolean removed = vsAuthOp.removeValueSet(new URI("VSD:AUTHORING:JUNIT:TEST1"), revInfo); </pre>

removeDefinitionEntry

This function provides capability to remove Definition Entry (rule set) of a Value Set.

removeDefinitionEntry(URI valueSetURI, Long ruleOrder, RevisionInfo revision)

Description:	Removes definition entry (rule set) of a value set definition.
Input:	java.net.URI valueSetURI - (Mandatory) URI of value set definition that contains the definition entry to be removed. java.lang.Long ruleOrder - (Mandatory) Rule set order id of a definition entry to be removed. org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	boolean - True; if remove was success
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	• <i>Step 1:</i> Instantiate ValueSetAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();</pre> • <i>Step 2:</i> Populate RevisionInfo object: <pre>RevisionInfo revInfo = new RevisionInfo(); revInfo.setChangeAgent("change Agent Name"); revInfo.setChangeInstruction("here goes the change Instructions"); revInfo.setDescription("description of the resource"); revInfo.setEditOrder(1L); revInfo.setRevisionDate(new Date()); revInfo.setRevisionId(UUID.randomUUID().toString());</pre> • <i>Step 3:</i> call remove definition entry method by passing the value set and rule order id information: <pre>boolean removed = vsAuthOp.removeDefinitionEntry(new URI("VSD:AUTHORING:JUNIT:TEST2"), 1L, revInfo);</pre>
---------------------	--

removeValueSetProperty

This function provides capability to remove property of a Value Set.

removeValueSetProperty(URI valueSetURI, String propertyId, RevisionInfo revision)'

Description:	Removes a property of a Value Set Definition.
Input:	• <i>java.net.URI valueSetURI</i> - (Mandatory) URI of value set definition that contains the property to be removed. • <i>java.lang.String propertyId</i> - (Mandatory) id of a property that needs to be removed. • <i>org.lexevs.cts2.core.update.RevisionInfo revision</i> - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	<i>boolean</i> - True; if remove was success
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	Step 1: Instantiate ValueSetAuthoringOperation if it is not done yet:
	<pre>org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();</pre>
	Step 2: Populate RevisionInfo object:
	<pre>RevisionInfo revInfo = new RevisionInfo(); revInfo.setChangeAgent("change Agent Name"); revInfo.setChangeInstruction("here goes the change Instructions"); revInfo.setDescription("description of the resource"); revInfo.setEditOrder(1L); revInfo.setRevisionDate(new Date()); revInfo.setRevisionId(UUID.randomUUID().toString());</pre>
	Step 3: Call remove property method by passing the value set and property information:
	<pre>boolean removed = vsAuthOp.removeValueSetProperty(new URI("VSD:AUTHORING:JUNIT:TEST1"), "propertyId1", revInfo); </pre>

Versionable Change Operations

Versionable Change operation provides capability to modify versionable attributes of Value Set like Status, Effective Date, Expiration Date, isActive and Owner.

updateValueSetStatus

This function modifies the status of a Value Set.

updateValueSetStatus(URI valueSetURI, String status, RevisionInfo revision)

Description:	Modifies the status of a Value Set.
Input:	- java.net.URI valueSetURI - (Mandatory) URI of value set definition. - java.lang.String status - (Mandatory) Modified status value. - org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	boolean - True; if update was success
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	Step 1: Instantiate ValueSetAuthoringOperation if it is not done yet:
	<pre>org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getValueSetAuthoringOperation();</pre>
	Step 2: Populate RevisionInfo object:
	<pre>RevisionInfo revInfo = new RevisionInfo(); revInfo.setChangeAgent("change Agent Name"); revInfo.setChangeInstruction("here goes the change Instructions"); revInfo.setDescription("description of the resource"); revInfo.setEditOrder(1L); revInfo.setRevisionDate(new Date()); revInfo.setRevisionId(UUID.randomUUID().toString());</pre>
	Step 3: Call status change method by passing the value set and status information:
	<pre>boolean changed = vsAuthOp.updateValueSetStatus(new URI("VSD:AUTHORING:JUNIT:TEST2"), "pending", revInfo);;</pre>

updateValueSetVersionable

This function provides capability to modify Value Set versionable attributes like effective date, expiration date, owner, status etc.

updateValueSetVersionable(URI valueSetURI, Versionable changedVersionable, RevisionInfo revision)

Description:	Update Value Set versionable attributes like effective date, expiration date, owner, status etc.
Input:	- java.net.URI valueSetURI - (Mandatory) URI of value set definition. - org.LexGrid.commonTypes.Versionable changedVersionable - (Mandatory) Modified versionable attributes (like:owner, effectiveDate, expirationDate, status etc). - org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	boolean - True; if update was success
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

**Sample
Call:**

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

```
org.lexevs.cts2.author.ValueSetAuthoringOperation vsAuthOp = LexEvsCTS2Impl.defaultInstance().  
getAuthoringOperation().getValueSetAuthoringOperation();
```

- *Step 2:* Populate RevisionInfo object:

```
RevisionInfo revInfo = new RevisionInfo();  
revInfo.setChangeAgent("change Agent Name");  
revInfo.setChangeInstruction("here goes the change Instructions");  
revInfo.setDescription("description of the resource");  
revInfo.setEditOrder(1L);  
revInfo.setRevisionDate(new Date());  
revInfo.setRevisionId(UUID.randomUUID().toString());
```

- *Step 3:* Populate versionable changes:

```
Versionable changedVersionable = new Versionable();  
changedVersionable.setEffectiveDate(new Date());  
changedVersionable.setIsActive(true);  
changedVersionable.setOwner("new Owner");  
changedVersionable.setStatus("new status");
```

- *Step 4:* Call change versionable method by passing the value set and changed versionable information:

```
boolean changed = vsAuthOp.updateValueSetVersionable(new URI("VSD:AUTHORING:JUNIT:TEST2"),  
changedVersionable, revInfo);
```