

LexEVS 6.0 CTS2 Authoring 2 - Code System Authoring Operation API

Contents of this Page

- [Introduction](#)
- [Interface](#)
- [Revision Information](#)
- [Authoring Functions](#)
 - [Create Operations](#)
 - [createCodeSystem](#)
 - [addCodeSystemProperties](#)
 - [createCodeSystemSuppliment](#)
 - [createConcept](#)
 - [addNewConceptProperty](#)
 - [createAssociationType](#)
 - [Edit Operations](#)
 - [updateCodeSystem](#)
 - [updateCodeSystemProperties](#)
 - [updateConcept](#)
 - [updateConceptProperty](#)
 - [updateAssociationType](#)
 - [Remove Operations](#)
 - [removeCodeSystem](#)
 - [removeCodeSystemProperty](#)
 - [deleteConcept](#)
 - [deleteConceptProperty](#)
 - [Status Change Operations](#)
 - [updateCodeSystemVersionStatus](#)
 - [updateConceptStatus](#)

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 Code System Authoring API provides capability to author Code System and its contents.

Here are the authoring functions that can be performed on code system and its contents:

- **Create** - This function provides capability to create:
 - New Code System
 - Add new property to Code System
 - New Concept
 - Add new property to Concept
 - New AssociationType
 - New Code System Suppliment
- **Edit** - This function provides capability to update:
 - Code System metadata
 - Property of a Code System
 - Concept metadata
 - Property of a Concept
 - AssociationType
- **Remove** - This function provides capability to remove:
 - Entire Code System
 - Property of a Code System
 - Concept
 - Property of a Concept
- **Status Change** - This function provides capability to update Status attributes:
 - Code System Status
 - Concept Status

Interface

org.lexevs.cts2.author.CodeSystemAuthoringOperation is the main interface for all the authoring operations against Code System. This interface can be accessed using main LexEVSCTS2 interface, like:

```
org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthOp = new org.lexevs.cts2.LexEvsCTS2Impl().
getAuthoringOperation().getCodeSystemAuthoringOperation();
```

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. These is done 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 CodeSystemAuthoringOperation interface.

Create Operations

Create Operation provides capability to create Code System, Concept, AssociationType, Property and Code System Supplement. 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.

createCodeSystem

This function provides capability to create a new Code System to contain a set of new coded concepts. The Code System is created by defining the set of meta-data properties that describe it. At this point there no concepts are added.

`createCodeSystem(RevisionInfo revision, String codingSchemeName, String codingSchemeURI, String formalName, String defaultLanguage, long approxNumConcepts, String representsVersion, List<String> localNameList, List<org.LexGrid.commonTypes.Source> sourceList, Text copyright, Mappings mappings)`

Description:	Creates new coding system.
Input:	• org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc. • java.lang.String codingSchemeName - (Mandatory) Code System Name. • java.lang.String codingSchemeURI - (Mandatory) Code System URI. • java.lang.String formalName - (Optional) Formal name of a Code System. • java.lang.String defaultLanguage - (Optional) Default language of Code System. • long approxNumConcepts - (Optional) Approximate number of concepts this Code System may contain. • java.lang.String representsVersion - (Mandatory) Initial version of the Code System. • java.util.List<java.lang.String> localNameList - (Optional) Any local name(s)/reference(s) for this Code System used within the Code System. • java.util.List<org.LexGrid.commonTypes.Source> sourceList - (Optional) Source(s) of this Code System. • org.LexGrid.commonTypes.Text copyright - (Optional) Information about rights held in and over the Code System. Typically, copyright information includes a statement about various property rights associated with the Code System, including intellectual property rights. • org.LexGrid.naming.Mappings mappings - (Mandatory) A list of all of the local identifiers and defining URI's that are used in the Code System.
Output:	org.LexGrid.codingSchemes.CodingScheme - Created Code System
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

**Sample
Call:**

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

```
org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.  
defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();
```

- *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 new code system meta data:

```
String codingSchemeURI = "urn:oid:11.11.0.99";  
String representsVersion = "1.0";  
  
String codingSchemeName = "New Coding Scheme";  
String formalName = "CTS 2 API Created Code System";  
String defaultLanguage = "en";  
Long approxNumConcepts = new Long(1);  
List<String> localNameList = Arrays.asList();  
  
org.LexGrid.commonTypes.Source source = new org.LexGrid.commonTypes.Source();  
source.setContent("source");  
List<org.LexGrid.commonTypes.Source> sourceList = Arrays.asList(source);  
  
Text copyright = new Text();  
org.LexGrid.naming.Mappings mappings = new org.LexGrid.naming.Mappings();  
  
org.LexGrid.naming.SupportedLanguage supportedLang = new org.LexGrid.naming.SupportedLanguage();  
  
supportedLang.setLocalId("en");  
supportedLang.setUri("URI_for_lang_en");  
  
mappings.addSupportedLanguage(supportedLang);
```

- *Step 4:* call `create` method to create the code system:

```
CodingScheme codeScheme = csAuthOp.createCodeSystem(revInfo, codingSchemeName, codingSchemeURI,  
formalName, defaultLanguage, approxNumConcepts, representsVersion, localNameList, sourceList,  
copyright, mappings);
```

addCodeSystemProperties

This function provides capability to add new properties to Code System.

`addCodeSystemProperties(RevisionInfo revision, String codingSchemeName, String codingSchemeURI, String representsVersion, Properties properties)`

Description:	Adds new properties to code system.
---------------------	-------------------------------------

Input:	• <i>org.lexevs.cts2.core.update.RevisionInfo revision</i> - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc. • <i>java.lang.String codingSchemeName</i> - (Mandatory) Code System Name. • <i>java.lang.String codingSchemeURI</i> - (Mandatory) Code System URI. • <i>java.lang.String representsVersion</i> - (Mandatory) Initial version of the Code System. • <i>org.LexGrid.commonTypes.Properties properties</i> - (Mandatory) List of one or more properties.
Output:	<i>org.LexGrid.codingSchemes.CodingScheme</i> - Updated Code System
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:

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

```
org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthOp = LexEvsCTS2Impl.  
defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();
```

- *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 properties to add:

```
Properties properties = new Properties();  
  
Presentation prop = new Presentation();  
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.setDataType("Text datatype");  
prop.setValue(text);  
prop.setDegreeOfFidelity("degreeOfFidelity");  
prop.setMatchIfNoContext(true);  
prop.setRepresentationalForm("representationalForm");=  
  
properties.addProperty(prop);  
  
Presentation prop2 = new Presentation();  
prop2.setPropertyId("propertyId2");  
prop2.setPropertyName("propertyName-2");  
prop2.setIsActive(false);  
prop2.setLanguage("english-2");  
prop2.setOwner("owner-2");  
prop2.setPropertyType(PropertyTypes.PROPERTY.name());  
Text text2 = new Text();  
text2.setContent("content-2");  
text2.setDataType("Text datatype-2");  
prop2.setValue(text2);  
prop2.setDegreeOfFidelity("degreeOfFidelity-2");  
prop2.setMatchIfNoContext(true);  
prop2.setRepresentationalForm("representationalForm-2");  
  
properties.addProperty(prop2);
```

- *Step 4:* call `add properties` method by passing the code system information and new set of properties:

```
CodingScheme codeScheme = csAuthOp.addCodeSystemProperties (revInfo, "New Coding Scheme",  
"urn:oid:11.11.0.99", "1.0", properties);
```

This function creates a new Code System Supplement as a container of a set of concepts and concept properties to be appended to a target code system. This function does not add the concepts and properties.

```
createCodeSystemSuppliment(AbsoluteCodingSchemeVersionReference parent, AbsoluteCodingSchemeVersionReference  
suppliment)
```

Description:	Creates a new Code System Supplement as a container of a set of concepts and concept properties to be appended to a target code system.
Input:	• <i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference parent</i> - (Mandatory) Target Code System Version. • <i>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference suppliment</i> - (Mandatory) Supplement Code System Version.
Output:	None
Exception:	<i>org.lexevs.cts2.exception.author.InvalidCodeSystemSupplementException</i>
Sample Call:	• <i>Step 1:</i> Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl. defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</pre> • <i>Step 2:</i> Populate target code system version reference: <pre>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference parent = new org.LexGrid. LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference() parent.setCodingSchemeURN("parentCodeSystemURN"); parent.setCodingSchemeVersion("1.0");</pre> • <i>Step 3:</i> Populate new suppliment code system version reference: <pre>org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference suppliment = new org. LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference() suppliment.setCodingSchemeURN("supplimentCodeSystemURN"); suppliment.setCodingSchemeVersion("1.0");</pre> • <i>Step 4:</i> Call create suppliment method by passing the parent and suppliment code system version: <pre>csAuthorOp.createCodeSystemSuppliment(parent, suppliment);</pre>

createConcept

This function creates a concept to be included in a Code System. The new concept is defined by the set of meta-data properties that describe it.

```
createConcept(String codingSchemeUri, String codeSystemVersion, String conceptCode, String namespace,  
RevisionInfo revision)
```

Description:	Creates new concept in a code system.
Input:	• <i>java.lang.String codingSchemeURI</i> - (Mandatory) URI of Code System that will contain this new concept. • <i>java.lang.String codeSystemVersion</i> - (Mandatory) Version of Code System that will contain this new concept. • <i>java.lang.String conceptCode</i> - (Mandatory) Code or identifier of the concept. • <i>java.lang.String namespace</i> - (Mandatory) namespace of the concept. • <i>org.lexevs.cts2.core.update.RevisionInfo revision</i> - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	None
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	• <i>Step 1:</i> Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 create concept method by passing code system version and concept information: <pre>csAuthOp.createConcept("NCIT", "10.0", "C12345", "NCIT", revInfo);</pre>
---------------------	--

addNewConceptProperty

This function provides capability to add a new property to a concept.

addNewConceptProperty(String codingSchemeUri, String codeSystemVersion, String conceptCode, String namespace, Property property, RevisionInfo revision)	
Description:	Adds a new property to a concept.
Input:	• <i>java.lang.String codingSchemeURI - (Mandatory)</i> URI of a Code System that contains the concept. • <i>java.lang.String codeSystemVersion - (Mandatory)</i> Version of a Code System that contains the concept. • <i>java.lang.String conceptCode - (Mandatory)</i> Concept code to which a new property will be added. • <i>java.lang.String namespace - (Mandatory)</i> Namespace of a concept to which a new property will be added. • <i>org.LexGrid.commonTypes.Property property - (Mandatory)</i> New property that will be added to the concept. • <i>org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory)</i> Contains revision information like unique RevisionId, change description, author information etc.
Output:	None
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	Step 1: Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 code system information, concept information and a new property: <pre>csAuthOp.addNewConceptProperty("urn:oid:11.11.0.99", "1.0", "C12345", "NCIT", prop, revInfo);</pre>

createAssociationType

This function provides capability to create a new association type, an instance of which may be used to link two concepts.

createAssociationType(String codingSchemeUri, String codeSystemVersion, String relationsContainerName, String associationName, String forwardName, String reverseName, Boolean isNavigable, Boolean isTransitive, RevisionInfo revision)

Description:	Creates new association type.
Input:	java.lang.String codingSchemeURI - (Mandatory) URI of a Code System that contains the new association type. java.lang.String codeSystemVersion - (Mandatory) Version of a Code System that contains the new association type. java.lang.String relationsContainerName - (Mandatory) Relation container name that contains the new association type. java.lang.String associationName - (Mandatory) Name or identifier of a new association type. java.lang.String forwardName - (Optional) The name or role that the "from" entity plays with respect to the "to" entry. Should be phrased in terms of the default language of the association and imply direction. java.lang.String reverseName - (Optional) The name or role that the "to" entity plays with respect to the "from" entry. Should be phrased in terms of the default language of the association and imply direction. java.lang.Boolean isNavigable - (Optional) True means that the reverse direction of the associaton is "navigable", meaning that it makes sense to represent the target to source side of the association. java.lang.Boolean isTransitive - (Optional) True means that association is transitive (r(a,b), r(b,c) -> r(a,c)). False means not transitive. If absent, transitivity is unknown or not applicable. org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	None
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	- Step 1: Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 create association type method by passing the code system, concept and new association type information: <pre>csAuthOp.createAssociationType("urn:oid:11.11.0.99", "1.0", "SCT-ICD9Mapping", "SameAs", "SameAs", "SameAs", true, true, revInfo);</pre>
---------------------	---

Edit Operations

The edit operation provides the capability to modify existing code system resources like code system metadata, property, concept, association type etc. 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.

updateCodeSystem

This function provides capability to modify the meta-data that describes the Code System. This method takes in any changes to the meta-data as a form of input parameters. If 'null' no changes will be performed on that particular attribute.

updateCodeSystem(RevisionInfo revision, String codingSchemeName, String codingSchemeURI, String formalName, String defaultLanguage, long approxNumConcepts, String representsVersion, List<String> localNameList, List<org.LexGrid.commonTypes.Source> sourceList, Text copyright, Mappings mappings)

Description:	Modify the meta-data that describes the Code System.
Input:	org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc. java.lang.String codingSchemeName - (Mandatory) Name of the Code System that is getting modified. java.lang.String codingSchemeURI - (Mandatory) URI of a Code System that is getting modified. java.lang.String formalName - (Optional) Modified formal name of a Code System. java.lang.String defaultLanguage - (Optional) Modified default language of Code System. long approxNumConcepts - (Optional) Modified approximate number of concepts this Code System may contain. java.lang.String representsVersion - (Mandatory) Version of the Code System that is getting modified. java.util.List<java.lang.String> localNameList - (Optional) Modified local name(s)/reference(s) for this Code System used within the Code System. java.util.List<org.LexGrid.commonTypes.Source> sourceList - (Optional) Modified source(s) of this Code System. org.LexGrid.commonTypes.Text copyright - (Optional) Modified information about rights held in and over the Code System. Typically, copyright information includes a statement about various property rights associated with the Code System, including intellectual property rights. org.LexGrid.naming.Mappings mappings - (Optional) Modified list of all of the local identifiers and defining URI's that are used in the Code System.
Output:	org.LexGrid.codingSchemes.CodingScheme - Modified Code System
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	- Step 1: Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 modified code system meta data: <pre>String codingSchemeURI = "urn:oid:11.11.0.99"; String representsVersion = "1.0"; String codingSchemeName = "New Coding Scheme"; String formalName = "CTS 2 API Modified Code System"; String defaultLanguage = "eng"; Long approxNumConcepts = new Long(1); List<String> localNameList = Arrays.asList(); org.LexGrid.commonTypes.Source source = new org.LexGrid.commonTypes.Source(); source.setContent("source 2"); List<org.LexGrid.commonTypes.Source> sourceList = Arrays.asList(source); Text copyright = new Text(); org.LexGrid.naming.Mappings mappings = new org.LexGrid.naming.Mappings(); org.LexGrid.naming.SupportedLanguage supportedLang = new org.LexGrid.naming.SupportedLanguage(); supportedLang.setLocalId("eng"); supportedLang.setUri("URI_for_lang_eng"); mappings.addSupportedLanguage(supportedLang);</pre> - Step 4: call update method to modify the code system: <pre>CodingScheme codeScheme = csAuthOp.updateCodeSystem(revInfo, codingSchemeName, codingSchemeURI, formalName, defaultLanguage, approxNumConcepts, representsVersion, localNameList, sourceList, copyright, mappings);</pre>
---------------------	--

updateCodeSystemProperties

This function provides capability to update properties of a Code System.

updateCodeSystemProperties(RevisionInfo revision, String codingSchemeName, String codingSchemeURI, String representsVersion, Properties properties)'

Description:	Updates properties of a code system.
Input:	org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc. java.lang.String codingSchemeName - (Mandatory) Name of the Code System. java.lang.String codingSchemeURI - (Mandatory) Code System URI. java.lang.String representsVersion - (Mandatory) Version of the Code System. org.LexGrid.commonTypes.Properties properties - (Mandatory) List of one or more modified properties.

Output:	<code>org.LexGrid.codingSchemes.CodingScheme</code> - Updated Code System
Exception:	<code>org.LexGrid.LexBIG.Exceptions.LBException</code>
Sample Call:	- Step 1: Instantiate <code>CodeSystemAuthoringOperation</code> if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</pre> - Step 2: Populate <code>RevisionInfo</code> 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 properties to modify. Just modifying language and setting <code>isActive</code> to true for property 'propertyId1': <pre>Properties properties = new Properties(); Presentation prop = new Presentation(); prop.setPropertyId("propertyId1"); prop.setPropertyName("propertyName"); prop.setIsActive(true); prop.setLanguage("eng"); properties.addProperty(prop);</pre> - Step 4: call <code>modify properties</code> method by passing the code system information and modified set of properties: <pre>CodingScheme codeScheme = csAuthOp.updateCodeSystemProperties(revInfo, "New Coding Scheme", "urn:oid:11.11.0.99", "1.0", properties);</pre>

updateConcept

This function provides capability to modify existing concept in a Code System.

`updateConcept(String codingSchemeUri, String codeSystemVersion, Entity entity, RevisionInfo revisionInfo)`

Description:	Modified an existing concept in a code system.
Input:	<code>java.lang.String codingSchemeURI</code> - (Mandatory) URI of Code System that contains the concept. <code>java.lang.String codeSystemVersion</code> - (Mandatory) Version of Code System that contains the concept. <code>org.LexGrid.concepts.Entity entity</code> - (Mandatory) Modified concept(Entity object). <code>org.lexevs.cts2.core.update.RevisionInfo revision</code> - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	None
Exception:	<code>org.LexGrid.LexBIG.Exceptions.LBException</code>

Sample Call:	Step 1: Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 Entity with modified concept: <pre>Entity entityToUpdate = new Entity(); entityToUpdate.setEntityCode("C12345"); entityToUpdate.setEntityCodeNamespace("NCIT"); entityToUpdate.setEntityDescription(Constructors.createEntityDescription("Modified ED"));</pre>
	Step 4: call modify concept method by passing code system version and concept information: <pre>csAuthOp.updateConcept("NCIT", "10.0", entityToUpdate, revInfo);</pre>

updateConceptProperty

This function provides capability to modify existing property of a concept.

updateConceptProperty(String codingSchemeUri, String codeSystemVersion, String conceptCode, String namespace, Property property, RevisionInfo revision)

Description:	Modifies existing property of a concept.
Input:	<i>java.lang.String codingSchemeURI</i> - (Mandatory) URI of a Code System that contains the concept. <i>java.lang.String codeSystemVersion</i> - (Mandatory) Version of a Code System that contains the concept. <i>java.lang.String conceptCode</i> - (Mandatory) Concept code that contains the property. <i>java.lang.String namespace</i> - (Mandatory) Namespace of a concept that contains the property. <i>org.LexGrid.commonTypes.Property property</i> - (Mandatory) Modified property. <i>org.lexevs.cts2.core.update.RevisionInfo revision</i> - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	None
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	Step 1: Instantiate CodeSystemAuthoringOperation if it is not done yet:
	<pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 code system information, concept information and the modified property:
	<pre>csAuthOp.updateConceptProperty("urn:oid:11.11.0.99", "1.0", "C12345", "NCIT", prop, revInfo);</pre>

updateAssociationType

This function provides capability to modify existing association type.

updateAssociationType(String codingSchemeUri, String codeSystemVersion, AssociationEntity associationEntity, RevisionInfo revision)

Description:	Modifies existing association type.
Input:	java.lang.String codingSchemeURI - (Mandatory) URI of a Code System that contains the association type. java.lang.String codeSystemVersion - (Mandatory) Version of a Code System that contains the association type. org.LexGrid.relations.AssociationEntity associationEntity - (Mandatory) Modified association type. org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	None
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	Step 1: Instantiate CodeSystemAuthoringOperation if it is not done yet:
	<pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 association type. Just modified isActive to 'true' and forwardName to 'newForwardName' for associationType 'SameAS':
	<pre>AssociationEntity associationEntity = new AssociationEntity(); associationEntity.setEntityCode("SameAs"); associationEntity.setIsActive(true); associationEntity.setForwardName("NewForwardName");</pre>
	Step 4: call update association type method by passing the code system information and the modified association type:
	<pre>csAuthOp.updateAssociationType("urn:oid:11.11.0.99", "1.0", associationEntity, revInfo);</pre>

Remove Operations

Remove operation provides capability to remove code system and its contents.

removeCodeSystem

This function provides capability to remove entire code system from terminology service.

removeCodeSystem(RevisionInfo revision, String codingSchemeURI, String representsVersion)

Description:	Remove entire code system from terminology service.
Input:	org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc. java.lang.String codingSchemeURI - (Mandatory) URI of a Code System that needs to be removed. java.lang.String codeSystemVersion - (Mandatory) Version of a Code System that needs to be removed.
Output:	boolean - True means deletion was success, otherwise false.
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	• <i>Step 1:</i> Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 code system method by passing the code system information: <pre>csAuthOp.removeCodeSystem(revInfo, "urn:oid:11.11.0.99", "1.0");</pre>
---------------------	---

removeCodeSystemProperty

This function provides capability to remove property of a Code System.

removeCodeSystemProperty(RevisionInfo revision, String codingSchemeURI, String representsVersion, String propertyId)

Description:	Removes a property of a Code System
Input:	• <i>org.lexevs.cts2.core.update.RevisionInfo revision</i> - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc. • <i>java.lang.String codingSchemeURI</i> - (Mandatory) Code System URI that contains the property. • <i>java.lang.String representsVersion</i> - (Mandatory) Version of the Code System that contains the property. • <i>java.lang.String propertyId</i> - (Mandatory) Identifier of a property that needs to be removed.
Output:	<i>org.LexGrid.codingSchemes.CodingScheme</i> - Updated Code System
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	• <i>Step 1:</i> Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 property method by passing the code system information and the property identifier: <pre>CodingScheme codeScheme = csAuthorOp.removeCodeSystemProperty(revInfo, "urn:oid:11.11.0.99", "1.0", "propertId1");</pre>
---------------------	--

deleteConcept

This function provides capability to remove a concept from code system.

```
deleteConcept(String codingSchemeUri, String codeSystemVersion, String conceptCode, String namespace, RevisionInfo revision)
```

Description:	Removes a concept from code system.
Input:	• <i>java.lang.String codingSchemeURI</i> - (Mandatory) URI of a Code System that contains the concept to be removed. • <i>java.lang.String codeSystemVersion</i> - (Mandatory) Version of a Code System that contains the concept to be removed. • <i>java.lang.String conceptCode</i> - (Mandatory) Identifier/Code of a Concept to be removed. • <i>java.lang.String namespace</i> - (Mandatory) Namespace of a Concept 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:	None
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	- Step 1: Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 concept method by passing the code system and concept information: <pre>csAuthOp.deleteConcept("urn:oid:11.11.0.99", "1.0", "C1234", 'NCIT", revInfo);</pre>
---------------------	--

deleteConceptProperty

This function provides capability to remove property of a concept.

`deleteConceptProperty(String codingSchemeUri, String codeSystemVersion, String conceptCode, String namespace, Property property, RevisionInfo revision)`

Description:	Removes a property of a concept.
Input:	<i>java.lang.String codingSchemeURI</i> - (Mandatory) Code System URI that contains the concept. <i>java.lang.String representsVersion</i> - (Mandatory) Version of the Code System that contains the concept. <i>java.lang.String conceptCode</i> - (Mandatory) Concept code that contains the property. <i>java.lang.String namespace</i> - (Mandatory) Namespace of a concept that contains the property. <i>org.LexGrid.commonType.Property property</i> - (Mandatory) 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:	None
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

Sample Call:	Step 1: Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 property that needs to be removed: <pre>Property propertyToRemove = new Property(); propertyToRemove.setPropertyId("p1");</pre> Step 4: Call remove property method by passing the code system, concept and property information: <pre>csAuthOp.deleteConceptProperty("urn:oid:11.11.0.99", "1.0", "C1234", "NCIT", propertyToRemove, revInfo);</pre>
---------------------	---

Status Change Operations

Status Change operations provides the capability to change the status a code system or a concept. Status could anything like: isActive, retired, published, pending etc. There is also a flag called 'isActive' that is associated with both code system and concept meta-data that can be changed using this function as well.

'isActive' flag: True means that this resource is searchable and browsable if the temporal context of the operation falls between effectiveDate and expirationDate. False means that this resource is only accessible if requested by id or by a search that specifies that inactive retrievals are allowed. Default: True

updateCodeSystemVersionStatus

This function modifies the status of a code system.

updateCodeSystemVersionStatus(String codingSchemeURI, String codeSystemVersion, String status, Boolean isActive, RevisionInfo revision)

Description:	Modifies the status of a code system.
Input:	java.lang.String codingSchemeURI - (Mandatory) Code System URI. java.lang.String representsVersion - (Mandatory) Version of the Code System. java.lang.String status - (Mandatory) Modified status value. java.lang.Boolean isActive - (Optional) Set to true or false. Null for no change. org.lexevs.cts2.core.update.RevisionInfo revision - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	None
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample Call:	• <i>Step 1:</i> Instantiate CodeSystemAuthoringOperation if it is not done yet: <pre>org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();</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 status change method by passing the code system and status information: <pre>csAuthOp.updateCodeSystemVersionStatus("urn:oid:11.11.0.99", "1.0", "Active", true, revInfo);</pre>
---------------------	--

updateConceptStatus

This function modifies the status of a concept.

updateConceptStatus(String codingSchemeUri, String codeSystemVersion, String conceptCode, String namespace,String status,Boolean isActive, RevisionInfo revisionInfo)

Description:	Modifies the status of a concept.
Input:	• <i>java.lang.String codingSchemeURI</i> - (Mandatory) Code System URI that contains the concept. • <i>java.lang.String representsVersion</i> - (Mandatory) Version of the Code System that contains the concept. • <i>java.lang.String conceptCode</i> - (Mandatory) Code/identifier of a concept. • <i>java.lang.String namespace</i> - (Mandatory) Mamespace of a concept. • <i>java.lang.String status</i> - (Mandatory) Modified status value. • <i>java.lang.Boolean isActive</i> - (Optional) Set to true or false. Null for no change. • <i>org.lexevs.cts2.core.update.RevisionInfo revision</i> - (Mandatory) Contains revision information like unique RevisionId, change description, author information etc.
Output:	None
Exception:	<i>org.LexGrid.LexBIG.Exceptions.LBException</i>

**Sample
Call:**

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

```
org.lexevs.cts2.author.CodeSystemAuthoringOperation csAuthorOp = LexEvsCTS2Impl.  
defaultInstance().getAuthoringOperation().getCodeSystemAuthoringOperation();
```

- *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:* Call status change method by passing the code system, concept and status information:

```
csAuthorOp.updateCodeSystemVersionStatus("urn:oid:11.11.0.99", "1.0", "C1234", "NCIT",  
"Active", true, revInfo);
```