

LexEVS 6.0 CTS2 Authoring 1 - Association Authoring Operation API

Contents of this Page

- [Introduction](#)
 - [Create](#)
 - [Status Change](#)
- [Interface](#)
- [Revision Information](#)
 - [org.LexGrid.versions.Revision](#)
 - [org.LexGrid.versions.EntryState](#)
- [Association Authoring Functions](#)
 - [Create Association Operations](#)
 - [Update Association Operations](#)

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 Associations in a Code System:

Create

This function provides capability to create:

- New Association
- Add Qualifiers to the Association
- Determine Source and Target Code Systems for the Association
- New Coding Scheme consisting of mappings of one system's concepts to another.
- New Mapping for an existing mapping scheme
- New Association for an existing coding scheme.

Status Change

This function provides capability to update Status attributes:

- Association Status

Interface

```
org.lexevs.cts2.LexEvsCTS2 cts2 = org.lexevs.cts2.LexEvsCTS2Impl.defaultInstance();
org.lexevs.cts2.author.AssociationAuthoringOperation associationAuthoring =
cts2.getAuthoringOperation().getAssociationAuthoringOperation();
```

Revision Information

Authoring requires information helping to determine whether the edited coding scheme element is new or revises an existing association. This requires not only a containing revision, but also a entry state object for each versionable element of the Association being created. A single object of each is passed into the association creating method providing mandatory information about this revision which in most use cases here is a new association.

org.LexGrid.versions.Revision

Revision 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

org.LexGrid.versions.EntryState

- **java.lang.String containingRevision** - (**Mandatory**)The revision that contains this particular entry state change
- **private java.lang.Long relativeOrder** - (Optional)The relative order that this state change should be applied within the context of the containing revision
- **private org.LexGrid.versions.types.ChangeType changeType** - (Context determined. NEW for new association)The type of change that occurred between this state and the previous.
- **private java.lang.String prevRevision** - (Optional)The unique identifier of the state of this entry was at prior to this change.

Association Authoring Functions

Create Association Operations

```
createAssociation(boolean createMappingScheme, Revision revision, EntryState entryState,
AbsoluteCodingSchemeVersionReference mappingScheme, AbsoluteCodingSchemeVersionReference
sourceCodeSystemIdentifier, AbsoluteCodingSchemeVersionReference targetCodeSystemIdentifier, String
sourceConceptCodeIdentifier, String targetConceptCodeIdentifier, String relationsContainerName, String
associationType, AssociationQualification[] associationQualifiers)
```

Description:	Creates new Association
Input:	<ul style="list-style-type: none"> • boolean createMappingScheme - create a mapping coding scheme if one does not exist • org.LexGrid.versions.Revision revision - create a mapping coding scheme if one does not exist • org.LexGrid.versions.EntryState entryState - revision data container granular to the versionable class. • Aorg.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference mappingScheme - existing mapping scheme, required if adding mapped association • org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference sourceCodeSystemIdentifier - minimum code system identification • org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference targetCodeSystemIdentifier - minimum code system identification • java.lang.String sourceConceptCodeIdentifier - source concept code • java.lang.String targetConceptCodeIdentifier - target concept code • java.lang.String relationsContainerName - relations container identifier • java.lang.String associationType - association type identifier • org.LexGrid.relations.AssociationQualification[] associationQualifiers - qualifications to add to this association
Output:	org.LexGrid.relations.AssociationSource - Created Association Structure
Exception:	org.LexGrid.LexBIG.Exceptions.LBException

Sample
Call:

- Step 1: Create Revision Elements:

```
Revision revision = new Revision(); - initialize Revision object
revision.setChangeAgent("Mayo Foundation"); - Set the change agent if desired
Text changeInstructions = new Text(); - initialize Text Object
changeInstructions.setContent("To be applied at next source release"); - Define change
instructions if desired.
revision.setChangeInstructions(changeInstructions); - Set change instructions
revision.setEditOrder(new Long(1)); - Set edit order
EntityDescription entityDescription = new EntityDescription(); - Initialize EntityDescription
Object
entityDescription.setContent("TestCTS2AssociationRevision"); - Set content for this object
revision.setEntityDescription(entityDescription); - Set revision entityDescription

EntryState entryState = new EntryState(); - Initialize EntryStateObject
entryState.setChangeType(ChangeType.NEW); - This is a new association -- set ChangeType to "NEW"
entryState.setContainingRevision("FirstRevision_12_09_2010"); - Define the revision identifier
entryState.setRelativeOrder(new Long(1)); - Set relative order
```

- Step 2: Define a source scheme:

```
AbsoluteCodingSchemeVersionReference sourceCodeSystemIdentifier = new
AbsoluteCodingSchemeVersionReference();
sourceCodeSystemIdentifier.setCodingSchemeURN("http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.
owl#");
sourceCodeSystemIdentifier.setCodingSchemeVersion(10.10a);
```

- Step 3: Define a target scheme:

```
AbsoluteCodingSchemeVersionReference targetCodeSystemIdentifier = new
AbsoluteCodingSchemeVersionReference();
targetCodeSystemIdentifier.setCodingSchemeURN("urn:oid:2.16.840.1.113883.6.2");
targetCodeSystemIdentifier.setCodingSchemeVersion(200909);
```

- Step 4: Name the unique identifier of the source concept:

```
String sourceConceptCodeIdentifier = "C27469";
```

- Step 5: Name the unique identifier of the target concept:

```
String targetConceptCodeIdentifier = "199";
```

- Step 6: Identify or name a new relations container

```
String relationsContainerName = "relations";
```

- Step 7: Name the association

```
String associationType = "SY";
```

- Step 8: call create method to create the association:
AssociationSource association = associationAuthoringOp.createAssociation(false, revision, entryState,
null, sourceCodeSystemIdentifier, targetCodeSystemIdentifier,
sourceConceptCodeIdentifier, targetConceptCodeIdentifier,
relationsContainerName, associationType, null);

Update Association Operations

```
updateAssociationStatus(Revision revision, EntryState entryState, AbsoluteCodingSchemeVersionReference scheme,
String relationsContainer, String associationName, String sourceCode, String sourceNamespace, String targetCode,
String targetNamespace, String instanceId, String status, boolean isActive)
```

Description:

Updates Association Status

Input:	<ul style="list-style-type: none"> org.LexGrid.versions.Revision revision - create a mapping coding scheme if one does not exist org.LexGrid.versions.EntryState entryState - revision data container granular to the versionable class. org.LexGrid.LexBIG.DataModel.Core.AbsoluteCodingSchemeVersionReference scheme - Scheme targeted for update. java.lang.String relationsContainerName - relations container identifier java.lang.String associationName - association identifier java.lang.String sourceCode - source concept code java.lang.String sourceNamespace - code system identification java.lang.String targetCode - target concept code java.lang.String targetNamespace - minimum code system identification java.lang.String instanceId - instance identifier for the association java.lang.String status - Status designation can be user defined boolean isActive - flag for Active status
Output:	boolean - flag for successful update
Exception:	org.LexGrid.LexBIG.Exceptions.LBException
Sample Call:	<ul style="list-style-type: none"> Step 1: Create Revision Elements: <div> <pre> Revision revision = new Revision(); - initialize Revision object revision.setChangeAgent("Mayo Foundation"); - Set the change agent if desired Text changeInstructions = new Text(); - initialize Text Object changeInstructions.setContent("To be applied at next source release"); - Define change instructions if desired. revision.setChangeInstructions(changeInstructions); - Set change instructions revision.setEditOrder(new Long(1)); - Set edit order EntityDescription entityDescription = new EntityDescription(); - Initialize EntityDescription Object entityDescription.setContent("TestCTS2AssociationRevision"); - Set content for this object revision.setEntityDescription(entityDescription); - Set revision entityDescription EntryState entryState = new EntryState(); - Initialize EntryStateObject entryState.setChangeType(ChangeType.MODIFY); - Association is being revised. Set ChangeType to "MODIFY" entryState.setContainingRevision("FirstRevision_12_09_2010"); - Define the revision identifier entryState.setRelativeOrder(new Long(1)); - Set relative order </pre> </div> Step 2: Define the scheme for the association update: <div> <pre> AbsoluteCodingSchemeVersionReference scheme = new AbsoluteCodingSchemeVersionReference(); scheme.setCodingSchemeURN("http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#"); scheme.setCodingSchemeVersion(10.10a); </pre> </div> Step 3: Name the unique identifier of the source concept: <div> <pre> String sourceCode = "C27469"; </pre> </div> Step 4: Define the namespace of the source concept supported for this coding scheme <div> <pre> String sourceNamespace = "NCI Thesaurus" </pre> </div> Step 5: Name the unique identifier of the target concept: <div> <pre> String targetCode = "C27469"; </pre> </div> Step 6: Identify or name a new relations container <div> <pre> String relationsContainerName = "relations"; </pre> </div> Step 7: Define the association <div> <pre> String associationName = "Disease_Has_Abnormal_Cell"; </pre> </div> Step 8: Identify the instance of this association to be updated. <div> <pre> String instanceID = "instance001"; </pre> </div>

- *Step 9:* Define the new status

```
String status = "PENDING_RETIREMENT";
```

- *Step 10:* Set isActive flag if necessary

```
boolean isActive = false;
```

- *Step 11:* call create method to create the association:
AssociationSource association = associationAuthoringOp.updateAssociationStatus(revision, entryState,
scheme, relationsContainerName, associationName, sourceCode,
namespace, targetCode, namespace, instanceId, status,
isActive);