caBIG® Platform Specific Model and Service Specification

LexEVS 6.0

Analytical Grid Services

Version 1.0

January 7, 2011

Authors	*Craig Stancl, Kevin Peterson, Scott Bauer, Sridhar Dwarkanath*
Editor	*Craig Stancl*
Reviewers	*Larry Brem*

Architecture Inception Team	*Craig Stancl, Harold Solbrig, Kevin Peterson, Scott Bauer, Sridhar Dwarkanath, Traci St. Martin*

DOCUMENT CHANGE HISTORY

Version Number	Date	Contributor	Description
1.0	**1/7/11**	**Craig Stancl, Sridhar Dwarkanath**	**Initial Draft**

Table of Contents

1 Executive Summary:

1.1 Service Description and Purpose

LexEVS 6.0 represents the next generation of NCI Enterprise Vocabulary Services. LexEVS is a mechanism for the standard storage of controlled vocabularies and ontologies defining a flexible format for accurately representing a wide variety of vocabularies and other lexically-based resources in several different server storage repositories as well as an XML format.

LexEVS provides a powerful and robust API and tool suite which permits access to controlled vocabulary content represented in the LexEVS model. This allows terminologies from a wide variety of resources such as RRF, OWL, and OBO to be represented and loaded to a single data base management system and accessed with a common set of tools and interfaces.

LexEVS is based off the LexGRID database schema and LexBIG API objects, where LexGRID defines how the terminologies are structured in the database and LexBIG defines how the terminology service looks as objects to the user. LexEVS provides optimizing query code that retrieves LexBIG objects, allowing the user to tailor calls to the terminology service in such a way that a discrete set of values is returned increasing utility and interoperability.

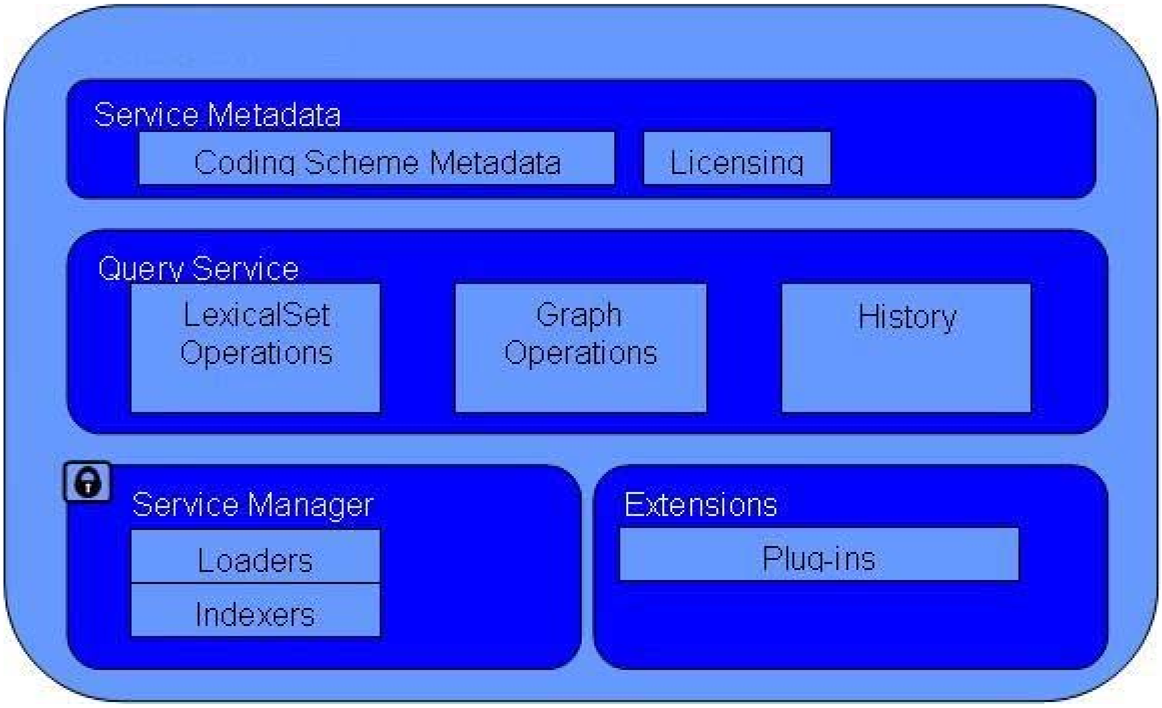
One of the requirements of LexEVS 6.0 is to align the LexEVS Analytical Grid Services component operations - including Search and Query Operations for Code Systems and Associations but excluding other LexEVS capabilities for querying and loading Value Sets, Concept Domains and Usage Contexts – to international efforts at developing common terminology service interfaces, specifically, the Health Level Seven (HL7) Common Terminology Services – Release 2 (CTS 2) standard.

NOTE: *For the purpose of this document, the terms “Code System” and “Coding Scheme” are synonymous.*

1.2 Scope

The scope of this PSM is constrained to the Analytical Grid Services components for LexEVS 6.0. Analytical Grid Services are those interfaces that are exposed on the Grid, and include the LexBIG domains of:

- LexBIGService – service identification interfaces
- CodedNodeGraph – A virtual graph where the edges represent associations and the nodes represent concept codes
- CodedNodeSet – A coded node set represents a flat list of coded entries
- HistoryService – Service reference to the history API servicing the given coding scheme
- LexBIGServiceMetadata – Interface to perform system-wide query over optionally loaded metadata for loaded code systems and providers



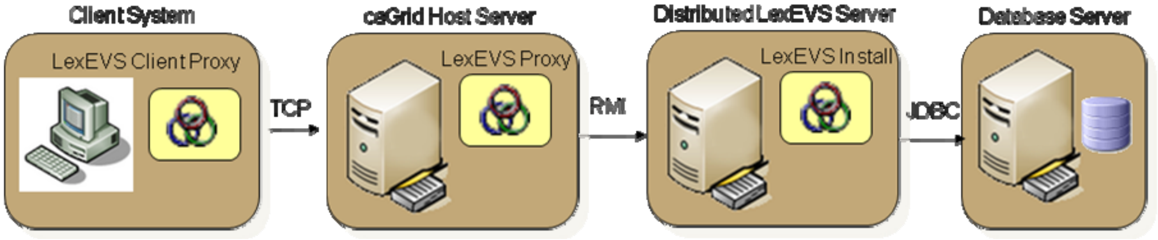
These service interfaces provide query and filtering support to the core LexBIGService interface, allowing code system content to be queried and grouped according to the different attributes and properties of code system content.

There are however, components of LexEVS that are purposely excluded from the Analytical Grid Services, such as terminology authoring and administration, value domain query and concept domain query. This section outlines the scope of LexEVS PSM with respect to the scope of the Analytical Grid Services.

1.3 Platform Details

LexEVS 6.0 Analytical Grid Services are built as a caGrid based service. Each of the defined grid analytical grid service operations (as defined later in the document) is viewable by the caGrid Portal (http://cagrid-portal.nci.nih.gov).

In the grid services environment, the client application makes calls the grid services interfaces which in turn call the distributed LexEVS API to access content in LexEVS. LexEVS for Analytical Grid Services consists of client system, caGrid Host Server, Distributed LexEVS server and database server. The client system is responsible for making calls to access controlled terminology content from the caGrid Host Server. The caGrid Host Server is responsible for routing information both to and from the client system from the LexEVS server. The LexEVS server is responsible for serving up structured terminology content represented in the LexGrid enabled repository (database server). Lastly, the database server houses the code systems available on LexEVS.



1.4 Referenced Standards

Listed are references to the standards and technical standards that the LexEVS Analytical Grid Service is using.

Domain Standards	Description
HL7 CTS 2	HL7's Common Terminology Services 2 (CTS 2) specification specifies functional model (CIM) outlining HL7's consensus requirement for terminology services. For the LexEVS Analytical Grid Services PSM, only the terminology and association query components of HL7 CTS 2 is considered to be in scope. LexEVS will ultimately implement much of the CTS 2 functionality, and as such, early identification of potential points of alignment is necessary.
ISO 21090 Health Informatics –	ISO 21090 data types provide a harmonized set of data type definitions for representing and exchanging healthcare related information.

Harmonized data types for information interchange	LexEVS 6.0 will interchange information using the 21090 data type specifications

Technology Standards	**Description**
SOAP 1.1	Simple Object Access Protocol ver. 1.1 is used to interact with the service

2 Conformance to Platform Independent Model

LexEVS 6.0 Analytical Grid Services PSM conforms to the corresponding PIM.

Platform Independent Model Name and Service Specification	**Platform Independent Model and Service Specification Version**	**Description & Link to the Platform Independent Model and Service Specification**
LexEVS 6.0 Platform Independent Model and Service Specification	1.5	https://wiki.nci.nih.gov/display/EVS/LexEVS+6.0+ECCF+Artifacts

2.1 Conformance Profile

This conformance profile defines the query capabilities for LexEVS coding scheme and service related data for LexEVS Analytical Grid Services. This profile is invoked when LexEVS Analytical Grid Services are called to query either service specific information or terminology content and return that content in ISO 21090 data types.

Conformance Profile No.	QS-CP2
Conformance Profile	LexEVS 21090 Full Query Conformance Profile

Name		
Functional Profiles	**Functional Profile No.**	**Functional Profile Name**
	QS-PF1	QS Service Query
	QS-PF2	QS Content Query
Semantic Profiles	**Semantic Profile No.**	**Semantic Profile Name**
	QS-SP1	CTS 2 Semantic Profile

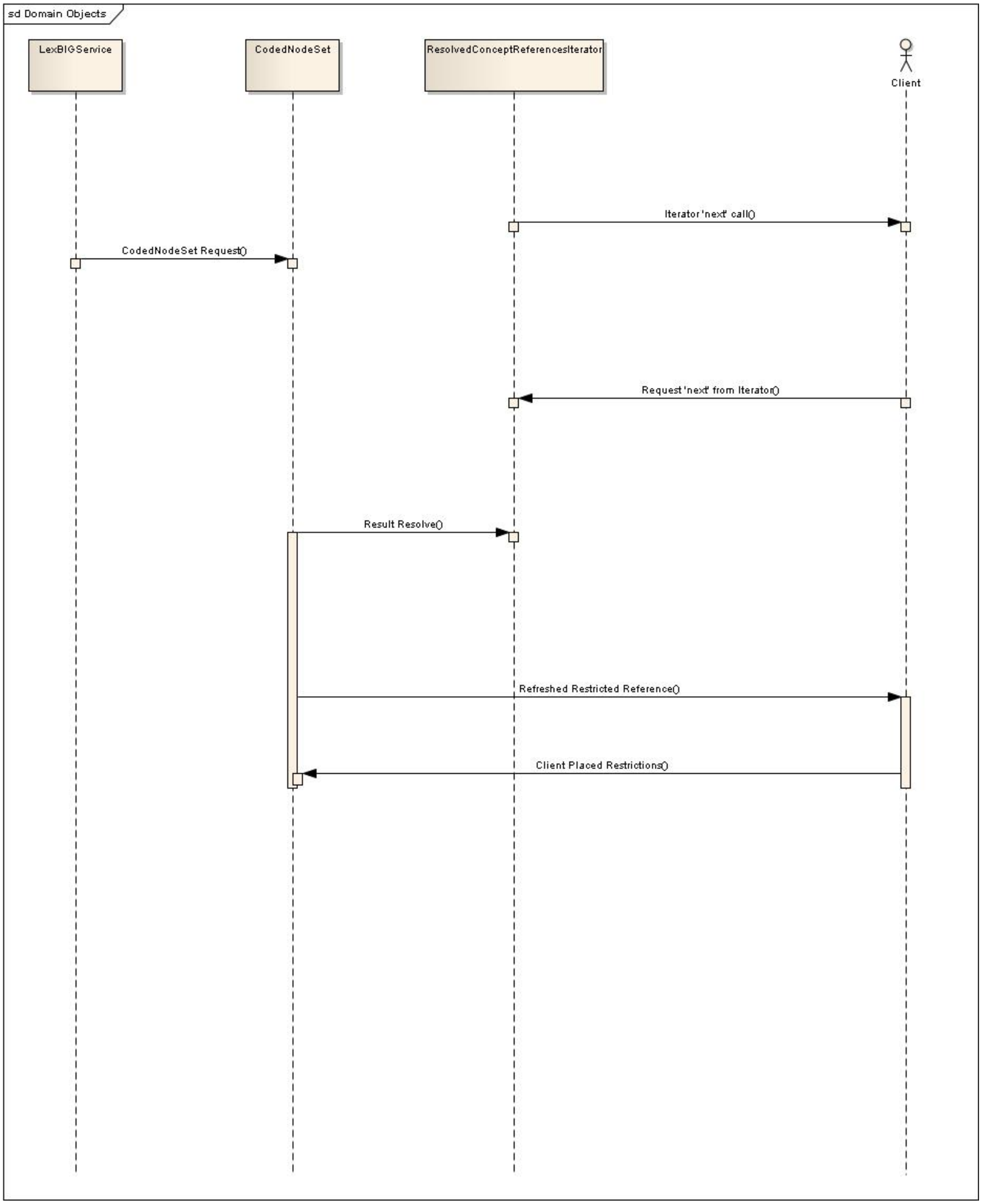
2.2 *Dynamic Interactions*

The LexEVS 6.0 Analytical Grid services depend on iterative interaction between service and client to build queries. As queries are initiated by the client, resources are held on the server in state to facilitate future access by the client. In this way the client may iteratively change the query, and execute when appropriate.

State is provided by WSRF (Web Services Resource Framework). Resources are held in state on the server and referenced by the client by unique ids. Clients may then interact directly with resources held in state on the server, as described below.

The Interaction below illustrates this Initiate -> Modify -> Resolve pattern.

1. The Client initiates a query to the service by making a request for a CodedNodeSet.
2. The client may make future calls to this CodedNodeSet.
3. The client at some point makes a request to Resolve the CodedNodeSet.


sd Domain Objects
LexBIGService
CodedNodeSet
ResolvedConceptReferencesIterator
Client
Iterator 'next' call()
CodedNodeSet Request()
Request 'next' from Iterator()
Result Resolve()
Refreshed Restricted Reference()
Client Placed Restrictions()

3 Platform Specific Model

3.1 Overview

Domain Model	Description
NCI's Implementation of ISO 21090 Data types	The service specifications will be based on NCI's restricted implementation of ISO 21090 data types

Technology	Affects
Globus Toolkit 4.0.3	Globus Toolkit is used to develop this grid service. This includes the WSRF Specification
JBoss 5.1	JBoss server will be used to deploy this grid service.

3.2 Assumptions and Dependencies

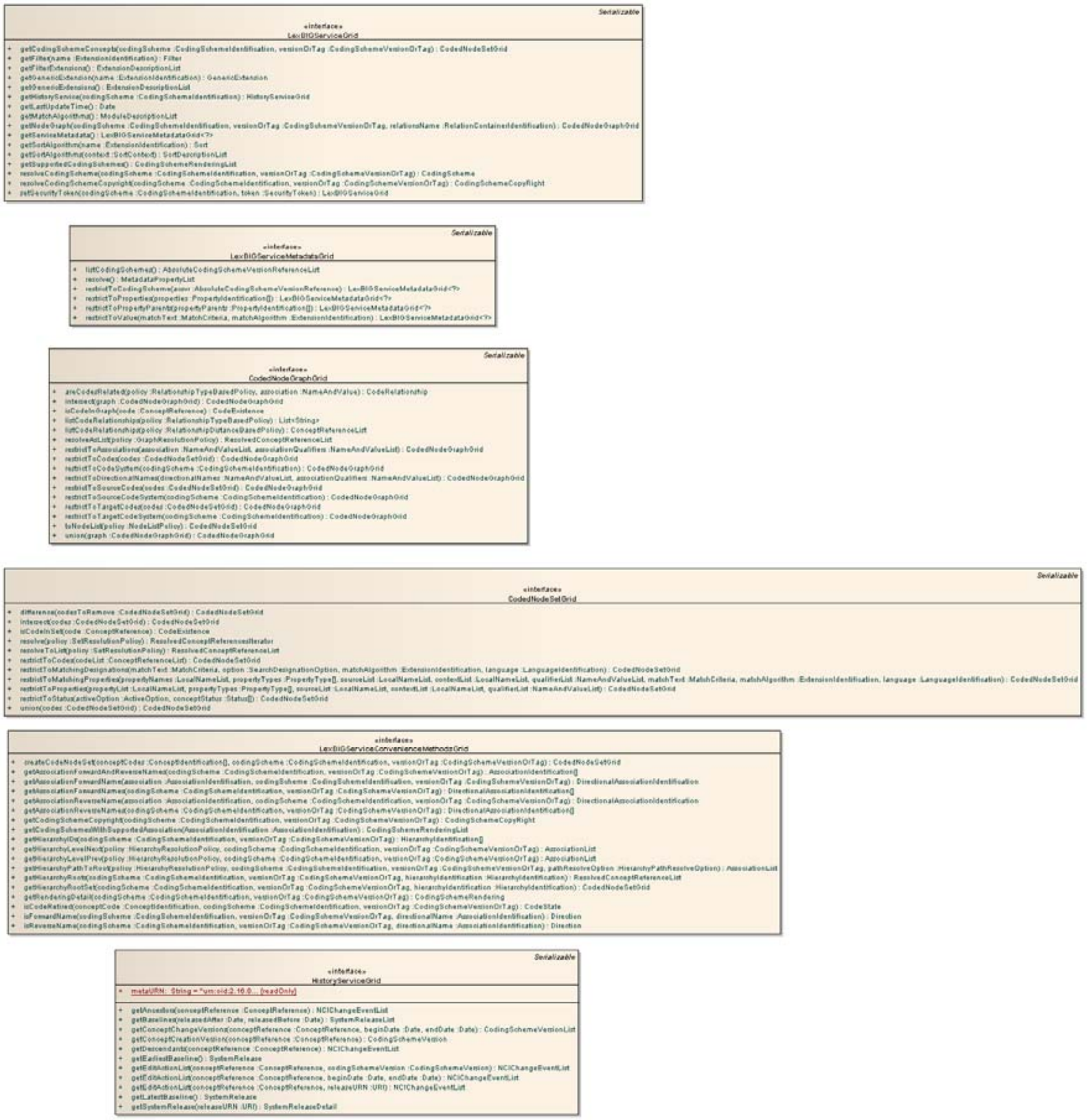
Assumptions	Affects
LexEVS 6.0 Analytical Grid Service will be deployed in its individual container	A new separate JBoss container needs to be procured for each of the analytical grid service installations
LexEVS 6.0 Analytical Grid Service will be deployed outside the NCI firewall.	Institutional Network team needs to ensure this is possible.

Dependency	Description
caGrid Production Environment	LexEVS 6.0 Analytical Grid Service relies on caGrid Production environment for advertisement and discovery

3.3 Service Interface

The LexEVS 6.0 Analytical Grid services are implemented as a Java API. The following link to the service UML model shows the main Java interfaces.

Links to: UML


Serializable
«interface»
LexBIGServiceGrid
+ getCodingSchemeConcepts(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : CodedNodeSetGrid
+ getFilter(name :ExtensionIdentification) : Filter
+ getFilterExtensions() : ExtensionDescriptionList
+ getGenericExtension(name :ExtensionIdentification) : GenericExtension
+ getGenericExtensions() : ExtensionDescriptionList
+ getHistoryService(codingScheme :CodingSchemeIdentification) : HistoryServiceGrid
+ getLastUpdateTime() : Date
+ getMatchAlgorithms() : ModuleDescriptionList
+ getNodeGraph(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag, relationsName :RelationContainerIdentification) : CodedNodeGraphGrid
+ getServiceMetadata() : LexBIGServiceMetadataGrid<?>
+ getSortAlgorithm(name :ExtensionIdentification) : Sort
+ getSortAlgorithms(context :SortContext) : SortDescriptionList
+ getSupportedCodingSchemes() : CodingSchemeRenderingList
+ resolveCodingScheme(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : CodingScheme
+ resolveCodingSchemeCopyright(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : CodingSchemeCopyRight
+ setSecurityToken(codingScheme :CodingSchemeIdentification, token :SecurityToken) : LexBIGServiceGrid
Serializable
«interface»
LexBIGServiceMetadataGrid
+ listCodingSchemes() : AbsoluteCodingSchemeVersionReferenceList
+ resolve() : MetadataPropertyList
+ restrictToCodingScheme(acsvr :AbsoluteCodingSchemeVersionReference) : LexBIGServiceMetadataGrid<?>
+ restrictToProperties(properties :PropertyIdentification[]) : LexBIGServiceMetadataGrid<?>
+ restrictToPropertyParents(propertyParents :PropertyIdentification[]) : LexBIGServiceMetadataGrid<?>
+ restrictToValue(matchText :MatchCriteria, matchAlgorithm :ExtensionIdentification) : LexBIGServiceMetadataGrid<?>
Serializable
«interface»
CodedNodeGraphGrid
+ areCodesRelated(policy :RelationshipTypeBasedPolicy, association :NameAndValue) : CodeRelationship
+ intersect(graph :CodedNodeGraphGrid) : CodedNodeGraphGrid
+ isCodeInGraph(code :ConceptReference) : CodeExistence
+ listCodeRelationships(policy :RelationshipTypeBasedPolicy) : List<String>
+ listCodeRelationships(policy :RelationshipDistanceBasedPolicy) : ConceptReferenceList
+ resolveAsList(policy :GraphResolutionPolicy) : ResolvedConceptReferenceList
+ restrictToAssociations(association :NameAndValueList, associationQualifiers :NameAndValueList) : CodedNodeGraphGrid
+ restrictToCodes(codes :CodedNodeSetGrid) : CodedNodeGraphGrid
+ restrictToCodeSystem(codingScheme :CodingSchemeIdentification) : CodedNodeGraphGrid
+ restrictToDirectionalNames(directionalNames :NameAndValueList, associationQualifiers :NameAndValueList) : CodedNodeGraphGrid
+ restrictToSourceCodes(codes :CodedNodeSetGrid) : CodedNodeGraphGrid
+ restrictToSourceCodeSystem(codingScheme :CodingSchemeIdentification) : CodedNodeGraphGrid
+ restrictToTargetCodes(codes :CodedNodeSetGrid) : CodedNodeGraphGrid
+ restrictToTargetCodeSystem(codingScheme :CodingSchemeIdentification) : CodedNodeGraphGrid
+ toNodeList(policy :NodeListPolicy) : CodedNodeSetGrid
+ union(graph :CodedNodeGraphGrid) : CodedNodeGraphGrid
Serializable
«interface»
CodedNodeSetGrid
+ difference(codesToRemove :CodedNodeSetGrid) : CodedNodeSetGrid
+ intersect(codes :CodedNodeSetGrid) : CodedNodeSetGrid
+ isCodeInSet(code :ConceptReference) : CodeExistence
+ resolve(policy :SetResolutionPolicy) : ResolvedConceptReferencesIterator
+ resolveToList(policy :SetResolutionPolicy) : ResolvedConceptReferenceList
+ restrictToCodes(codeList :ConceptReferenceList) : CodedNodeSetGrid
+ restrictToMatchingDesignations(matchText :MatchCriteria, option :SearchDesignationOption, matchAlgorithm :ExtensionIdentification, language :LanguageIdentification) : CodedNodeSetGrid
+ restrictToMatchingProperties(propertyNames :LocalNameList, propertyTypes :PropertyType[], sourceList :LocalNameList, contextList :LocalNameList, qualifierList :NameAndValueList, matchText :MatchCriteria, matchAlgorithm :ExtensionIdentification, language :LanguageIdentification) : CodedNodeSetGrid
+ restrictToProperties(propertyList :LocalNameList, propertyTypes :PropertyType[], sourceList :LocalNameList, contextList :LocalNameList, qualifierList :NameAndValueList) : CodedNodeSetGrid
+ restrictToStatus(activeOption :ActiveOption, conceptStatus :Status[]) : CodedNodeSetGrid
+ union(codes :CodedNodeSetGrid) : CodedNodeSetGrid
«interface»
LexBIGServiceConvenienceMethodsGrid
+ createCodeNodeSet(conceptCodes :ConceptIdentification[], codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : CodedNodeSetGrid
+ getAssociationForwardAndReverseNames(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : AssociationIdentification[]
+ getAssociationForwardName(association :AssociationIdentification, codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : DirectionalAssociationIdentification
+ getAssociationForwardNames(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : DirectionalAssociationIdentification[]
+ getAssociationReverseName(association :AssociationIdentification, codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : DirectionalAssociationIdentification
+ getAssociationReverseNames(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : DirectionalAssociationIdentification[]
+ getCodingSchemeCopyright(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : CodingSchemeCopyRight
+ getCodingSchemesWithSupportedAssociation(AssociationIdentification :AssociationIdentification) : CodingSchemeRenderingList
+ getHierarchyIDs(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : HierarchyIdentification[]
+ getHierarchyLevelNext(policy :HierarchyResolutionPolicy, codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : AssociationList
+ getHierarchyLevelPrev(policy :HierarchyResolutionPolicy, codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : AssociationList
+ getHierarchyPathToRoot(policy :HierarchyResolutionPolicy, codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag, pathResolveOption :HierarchyPathResolveOption) : AssociationList
+ getHierarchyRoots(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag, hierarchyIdentification :HierarchyIdentification) : ResolvedConceptReferenceList
+ getHierarchyRootSet(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag, hierarchyIdentification :HierarchyIdentification) : CodedNodeSetGrid
+ getRenderingDetail(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : CodingSchemeRendering
+ isCodeRetired(conceptCode :ConceptIdentification, codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : CodeState
+ isForwardName(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag, directionalName :AssociationIdentification) : Direction
+ isReverseName(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag, directionalName :AssociationIdentification) : Direction
Serializable
«interface»
HistoryServiceGrid
+ metaURN: String = "urn:oid:2.16.0... {readOnly}
+ getAncestors(conceptReference :ConceptReference) : NCIChangeEventList
+ getBaselines(releasedAfter :Date, releasedBefore :Date) : SystemReleaseList
+ getConceptChangeVersions(conceptReference :ConceptReference, beginDate :Date, endDate :Date) : CodingSchemeVersionList
+ getConceptCreationVersion(conceptReference :ConceptReference) : CodingSchemeVersion
+ getDescendants(conceptReference :ConceptReference) : NCIChangeEventList
+ getEarliestBaseline() : SystemRelease
+ getEditActionList(conceptReference :ConceptReference, codingSchemeVersion :CodingSchemeVersion) : NCIChangeEventList
+ getEditActionList(conceptReference :ConceptReference, beginDate :Date, endDate :Date) : NCIChangeEventList
+ getEditActionList(conceptReference :ConceptReference, releaseURN :URI) : NCIChangeEventList
+ getLatestBaseline() : SystemRelease
+ getSystemRelease(releaseURN :URI) : SystemReleaseDetail

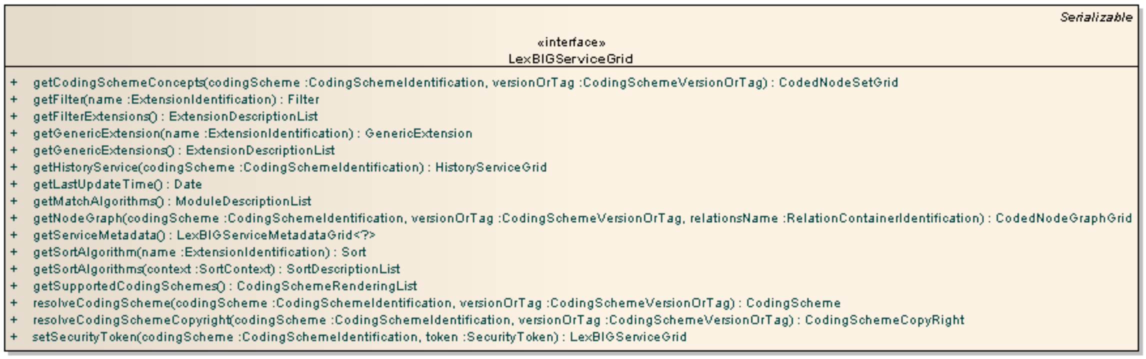
3.3.1 Interface Model

Implemented Interface No	Supported Interface Name	Interface Description	Link

LE_I_01	LexBIGServiceGrid	This interface represents the core interface to a LexBIG service.	JavaDoc
LE_I_02	CodedNodeSetGrid	A coded node set represents a flat list of coded entries.	JavaDoc
LE_I_03	CodedNodeGraphGrid	A virtual graph where the edges represent associations and the nodes represent coded entries. A CodedNodeGraph describes a graph that can be combined with other graphs, queried or resolved into an actual graph rendering.	JavaDoc
LE_I_05	LexBIGServiceMetadataGrid	Interface to perform system-wide query over metadata for loaded code systems and providers.	JavaDoc
LE_I_06	HistoryServiceGrid	The history service returns information about the change history of a coding scheme.	JavaDoc

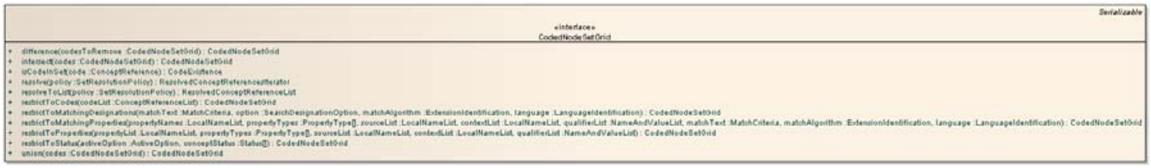
3.3.1.1 UML Diagram – Interface LexBIGServiceGrid (LE_I_01)

Link to: UML


Serializable
«interface»
LexBIGServiceGrid
+ getCodingSchemeConcepts(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : CodedNodeSetGrid
+ getFilter(name :ExtensionIdentification) : Filter
+ getFilterExtensions() : ExtensionDescriptionList
+ getGenericExtension(name :ExtensionIdentification) : GenericExtension
+ getGenericExtensions() : ExtensionDescriptionList
+ getHistoryService(codingScheme :CodingSchemeIdentification) : HistoryServiceGrid
+ getLastUpdateTime() : Date
+ getMatchAlgorithms() : ModuleDescriptionList
+ getNodeGraph(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag, relationsName :RelationContainerIdentification) : CodedNodeGraphGrid
+ getServiceMetadata() : LexBIGServiceMetadataGrid<?>
+ getSortAlgorithm(name :ExtensionIdentification) : Sort
+ getSortAlgorithms(context :SortContext) : SortDescriptionList
+ getSupportedCodingSchemes() : CodingSchemeRenderingList
+ resolveCodingScheme(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : CodingScheme
+ resolveCodingSchemeCopyright(codingScheme :CodingSchemeIdentification, versionOrTag :CodingSchemeVersionOrTag) : CodingSchemeCopyRight
+ setSecurityToken(codingScheme :CodingSchemeIdentification, token :SecurityToken) : LexBIGServiceGrid

3.3.1.2 UML Diagram – Interface CodedNodeSetGrid (LE_I_02)

Link to: UML

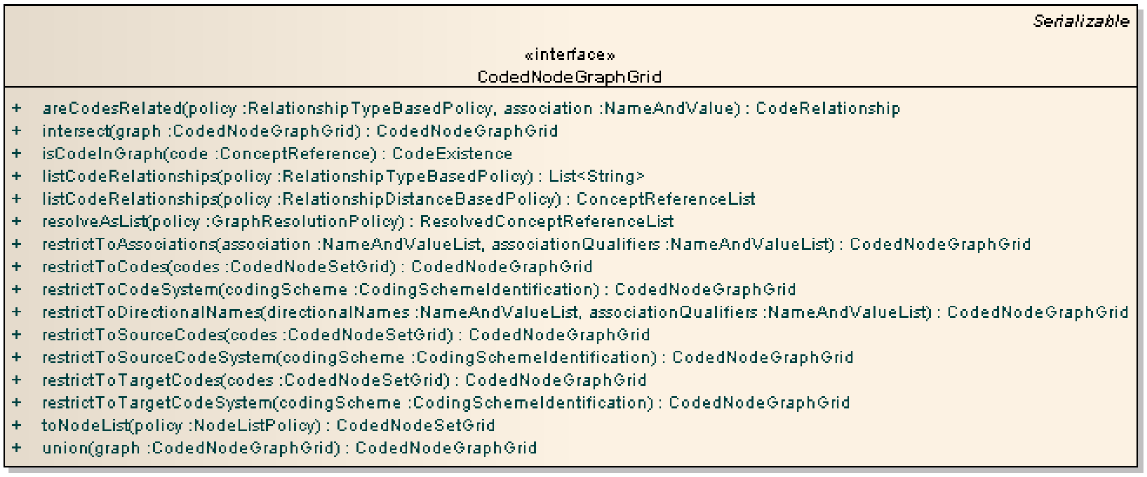


```
Serializable
«interface»
CodedNodeSetGrid

+ difference(codesToRemove :CodedNodeSetGrid) : CodedNodeSetGrid
+ intersect(codes :CodedNodeSetGrid) : CodedNodeSetGrid
+ isCodeInSet(code :ConceptReference) : CodeExistence
+ resolve(policy :SetResolutionPolicy) : ResolvedConceptReferenceIterator
+ resolveToList(policy :SetResolutionPolicy) : ResolvedConceptReferenceList
+ restrictToCodes(codeList :ConceptReferenceList) : CodedNodeSetGrid
+ restrictToMatchingDesignations(matchText :MatchCriteria, option :SearchDesignationOption, matchAlgorithm :ExtensionIdentification, language :LanguageIdentification) : CodedNodeSetGrid
+ restrictToMatchingProperties(propertyNames :LocalNameList, propertyTypes :PropertyType[], sourceList :LocalNameList, contextList :LocalNameList, qualifierList :NameAndValueList, matchText :MatchCriteria, matchAlgorithm :ExtensionIdentification, language :LanguageIdentification) : CodedNodeSetGrid
+ restrictToProperties(propertyList :LocalNameList, propertyTypes :PropertyType[], sourceList :LocalNameList, contextList :LocalNameList, qualifierList :NameAndValueList) : CodedNodeSetGrid
+ restrictToStatus(activeOption :ActiveOption, conceptStatus :Status[]) : CodedNodeSetGrid
+ union(codes :CodedNodeSetGrid) : CodedNodeSetGrid
```

3.3.1.3 UML Diagram – Interface CodedNodeGraphGrid (LE_I_03)

Link to: UML

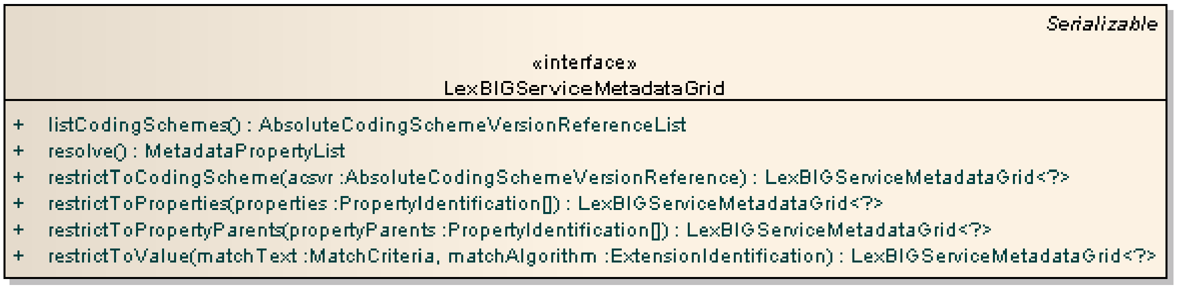


```
Serializable
«interface»
CodedNodeGraphGrid

+ areCodesRelated(policy :RelationshipTypeBasedPolicy, association :NameAndValue) : CodeRelationship
+ intersect(graph :CodedNodeGraphGrid) : CodedNodeGraphGrid
+ isCodeInGraph(code :ConceptReference) : CodeExistence
+ listCodeRelationships(policy :RelationshipTypeBasedPolicy) : List<String>
+ listCodeRelationships(policy :RelationshipDistanceBasedPolicy) : ConceptReferenceList
+ resolveAsList(policy :GraphResolutionPolicy) : ResolvedConceptReferenceList
+ restrictToAssociations(association :NameAndValueList, associationQualifiers :NameAndValueList) : CodedNodeGraphGrid
+ restrictToCodes(codes :CodedNodeSetGrid) : CodedNodeGraphGrid
+ restrictToCodeSystem(codingScheme :CodingSchemeIdentification) : CodedNodeGraphGrid
+ restrictToDirectionalNames(directionalNames :NameAndValueList, associationQualifiers :NameAndValueList) : CodedNodeGraphGrid
+ restrictToSourceCodes(codes :CodedNodeSetGrid) : CodedNodeGraphGrid
+ restrictToSourceCodeSystem(codingScheme :CodingSchemeIdentification) : CodedNodeGraphGrid
+ restrictToTargetCodes(codes :CodedNodeSetGrid) : CodedNodeGraphGrid
+ restrictToTargetCodeSystem(codingScheme :CodingSchemeIdentification) : CodedNodeGraphGrid
+ toNodeList(policy :NodeListPolicy) : CodedNodeSetGrid
+ union(graph :CodedNodeGraphGrid) : CodedNodeGraphGrid
```

3.3.1.4 UML Diagram – Interface LexBIGServiceMetadataGrid (LE_I_05)

Link to: UML

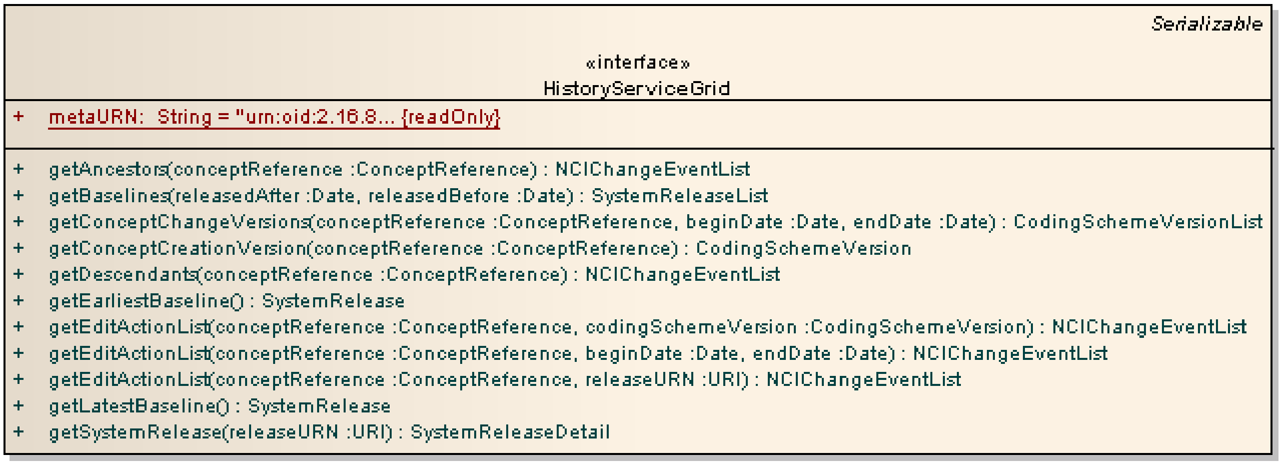


```
Serializable
«interface»
LexBIGServiceMetadataGrid

+ listCodingSchemes() : AbsoluteCodingSchemeVersionReferenceList
+ resolve() : MetadataPropertyList
+ restrictToCodingScheme(acsvr :AbsoluteCodingSchemeVersionReference) : LexBIGServiceMetadataGrid<?>
+ restrictToProperties(properties :PropertyIdentification[]) : LexBIGServiceMetadataGrid<?>
+ restrictToPropertyParents(propertyParents :PropertyIdentification[]) : LexBIGServiceMetadataGrid<?>
+ restrictToValue(matchText :MatchCriteria, matchAlgorithm :ExtensionIdentification) : LexBIGServiceMetadataGrid<?>
```

3.3.1.5 UML Diagram – Interface HistoryServiceGrid (LE_I_06)

Link to: UML



```
Serializable
«interface»
HistoryServiceGrid
+ metaURN: String = "urn:oid:2.16.8... {readOnly}
+ getAncestors(conceptReference :ConceptReference) : NCIChangeEventList
+ getBaselines(releasedAfter :Date, releasedBefore :Date) : SystemReleaseList
+ getConceptChangeVersions(conceptReference :ConceptReference, beginDate :Date, endDate :Date) : CodingSchemeVersionList
+ getConceptCreationVersion(conceptReference :ConceptReference) : CodingSchemeVersion
+ getDescendants(conceptReference :ConceptReference) : NCIChangeEventList
+ getEarliestBaseline() : SystemRelease
+ getEditActionList(conceptReference :ConceptReference, codingSchemeVersion :CodingSchemeVersion) : NCIChangeEventList
+ getEditActionList(conceptReference :ConceptReference, beginDate :Date, endDate :Date) : NCIChangeEventList
+ getEditActionList(conceptReference :ConceptReference, releaseURN :URI) : NCIChangeEventList
+ getLatestBaseline() : SystemRelease
+ getSystemRelease(releaseURN :URI) : SystemReleaseDetail
```

3.3.2 Operations Details for LexBIGServiceGrid

This interface represents the core interface to a LexEVS service.

Links to: JavaDoc and UML

3.3.2.1 getCodingSchemeConcepts

Returns the set of all concepts in the specified coding scheme.

Links to: JavaDoc and Schema XSD

Behavior Description	• The starting point for Lexical based matching and queries • Validation will ensure that: o The Requested Coding Scheme exists in the System o The specified version of the Coding Scheme is available o The specified Coding Scheme is Active • The returned CodedNodeSet is the starting point for building a lexical query.
Pre-Conditions	• The requested CodingScheme is available and Active in the System.
Inputs	• codingScheme - The local name, URI, or formal name of the requested CodingScheme • versionOrTag - A LexGrid:CodingSchemeVersionOrTag indicating the Version or Tag of the Requested CodingScheme. o If not provided, and there exists only one CodingScheme in the system matching the 'codingScheme' input, that CodingScheme will be

	used. o If more than one CodingScheme in the system matches the 'codingScheme' input, the CodingScheme tagged as "PRODUCTION" will be used.
Outputs	• A CodedNodeSetGrid, the starting point for lexical queries.
Post-Conditions	• A CodedNodeSetGrid for the requested CodingScheme is initialized
Exception Conditions	• The requested CodingScheme is not available in the system • The requested CodingScheme is not Active • There are more than one CodingScheme matching the 'codingScheme' input, and no 'versionOrTag' input is provided
Note	• The resulting CodedNodeSetGrid will include codes only of 'concept' type. For all other types, see the operation 'getNodeSet'

3.3.2.2 getFilter

Returns an instance of the filter extension registered with the given name

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns a Filter Extension, which may be used to restrict a set of results • Validation is carried out to ensure that: o A Filter Extension with the given name is available and registered correctly in the system • If all conditions are met, the Filter Extension is returned to the consumer
Pre-Conditions	• The Filter must have been properly registered as an Extension in the system
Inputs	• name – The Extension name of the requested Filter
Outputs	• Filter – an instance of the filter extension.
Post-Conditions	• The Filter Extension will be initialized and returned to the user, given the state and logic provided by the author of the Extension itself.
Exception Conditions	• The requested Filter Extension does not exist, or has not been registered properly.

3.3.2.3 getFilterExtensions

Returns a description of all registered extensions used to provide additional filtering of query results.

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns a list of Filter Extension that have been registered in the system
Pre-Conditions	• Any Filter Extensions must be correctly registered as Extensions to be listed
Inputs	• none
Outputs	• An ExtensionDescriptionList, which is a list describing each Filter Extension registered within the system.
Post-Conditions	• The populated ExtensionDescriptionList will be available to the user, and each of the Extensions listed will be available to the user.
Exception Conditions	• none

3.3.2.4 getGenericExtension

Returns an instance of the application-specific extension registered with the given name.

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns an application-specific Generic Extension that has been registered in the system • Validation is carried out to ensure that: o A Generic Extension with the given name is available and registered correction in the system • If all conditions are met, the Generic Extension is returned to the consumer
Pre-Conditions	• Any Generic Extensions must be correctly registered as Extensions to be available during the execution of this operation
Inputs	• name – the name of the requested Extension as it has be registered in the system
Outputs	• GenericExtension – an instance of the application-specific extension.

Post-Conditions	• The Generic Extension will be initialized and returned to the user, given the state and logic provided by the author of the Extension itself.
Exception Conditions	• The given 'name' input is: o Not registered correction in the system o Does not exist in the system

3.3.2.5 getGenericExtensions

Returns a description of all registered extensions used to implement application-specific behavior that is centrally accessible from a LexBIGService.

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns a list of Generic Extensions that have been registered in the system • Note: Only Generic Extensions (base class GenericExtension) will be listed here. All other classes are retrievable at the appropriate interface point (filter, sort, etc).
Pre-Conditions	• Any Generic Extensions must be correctly registered as Extensions to be listed
Inputs	• none
Outputs	• An ExtensionDescriptionList, which is a list describing each Filter Extension registered within the system.
Post-Conditions	• The populated ExtensionDescriptionList will be available to the user, and each of the Extensions listed will be available to the user.
Exception Conditions	• none

3.3.2.6 getHistoryService

Resolve a reference to the History Interface servicing the given coding scheme.

Links to: JavaDoc and Schema XSD

Behavior Description	• The entry point to the History Interface, given a requested CodingScheme • Validation is carried out to ensure that: o There has been History content loaded for the requested CodingScheme

	- If all conditions are met, a reference to the History Interface is returned to the consumer
Pre-Conditions	- History content must have been loaded and registered in the system for the requested CodingScheme
Inputs	- codingScheme – the localName or URI of the requested CodingScheme
Outputs	- HistoryServiceGrid - a reference to the History Interface for the requested CodingScheme
Post-Conditions	- The History Interface returned to the user will be initialized for the requested CodingScheme
Exception Conditions	- If the no History exists for the requested CodingScheme

3.3.2.7 getLastUpdateTime

Return the last time that the content of this service was changed

Links to: JavaDoc and Schema XSD

Behavior Description	- Returns the time of the last content change of the system. A content change consists of any change to the vocabulary metadata or the vocabulary itself in the system. - Note: A Tag assignment does not constitute a content change
Pre-Conditions	- History content must have been loaded and registered in the system for the requested CodingScheme
Inputs	- none
Outputs	- Date - the time of the last content change - Note: if the system has had no registered content changes, nothing will be returned
Post-Conditions	- None
Exception Conditions	- The system has not been initialized correctly

3.3.2.8 getMatchAlgorithms

Returns the full description of all supported match algorithms.

Links to: JavaDoc and Schema XSD

Behavior	- A description of all Match Algorithms (if any) is returned.

Description	
Pre-Conditions	• Any Match Algorithms must be available and registered in the system to be returned
Inputs	• none
Outputs	• ModuleDescriptionList - the full description of all Match Algorithms in the system.
Post-Conditions	• none
Exception Conditions	• The system has not been initialized correctly

3.3.2.9 getNodeGraph

Returns the node graph as represented in the particular relationship set in the coding scheme

Links to: JavaDoc and Schema XSD

Behavior Description	• The starting point for relation based matching and queries • Validation will ensure that: o The Requested Coding Scheme exists in the System o The specified version of the Coding Scheme is available o The specified Coding Scheme is Active • The returned NodeGraph is the starting point for building a relation based query.
Pre-Conditions	• The requested CodingScheme is available and Active in the System.
Inputs	• codingScheme - The local name, URI, or formal name of the requested CodingScheme • versionOrTag - A LexGrid:CodingSchemeVersionOrTag indicating the Version or Tag of the Requested CodingScheme. o If not provided, and there exists only one CodingScheme in the system matching the 'codingScheme' input, that CodingScheme will be used. o If more than one CodingScheme in the system matches the 'codingScheme' input, the CodingScheme tagged as "PRODUCTION" will be used. • relationsName - The name of the relations container to

	reference when generating the graph. If omitted, all native relation containers for the code system will be queried. Note: a 'native' container contains a set of associations defined by the coding scheme curators.
Outputs	• A CodedNodeGraphGrid, the starting point for relation queries.
Post-Conditions	• A CodedNodeGraphGrid for the requested CodingScheme is initialized
Exception Conditions	• The requested CodingScheme is not available in the system • The requested CodingScheme is not Active • There are more than one CodingScheme matching the ‘codingScheme’ input, and no ‘versionOrTag’ input is provided
Notes	• A 'native' container contains a set of associations defined by the coding scheme curators.

3.3.2.10 getServiceMetadata

Return an interface to perform system-wide query over metadata for loaded code systems and providers.

Links to: JavaDoc and Schema XSD

Behavior Description	• The entry point to the LexBIGServiceMetadata Interface
Pre-Conditions	• The LexEVS system is correctly initialized • Any Service Metadata has be loaded and registered to the system
Inputs	• none
Outputs	• LexBIGServiceMetadataGrid - a reference to the LexBIGServiceMetadataGrid Interface
Post-Conditions	• The LexBIGServiceMetadataGrid Interface is returned to the user will be initialized for the underlying system, with all loaded and registered metadata available for queries
Exception Conditions	• Exceptions will occur if: o The underlying system fails to initialize o The LexBIGServiceMetadataGrid Interface fails to initialize o There has be invalid metadata loaded to the system

3.3.2.11 getSortAlgorithm

Returns an instance of the sort extension registered with the given name

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns a Sort Interface, which may be used to sort a set of results • Validation is carried out to ensure that: o A Sort algorithm with the given name is available and registered in the system • If all conditions are met, the Sort Algorithm is returned to the consumer
Pre-Conditions	• Any Sort Algorithms must be available and registered in the system to be returned
Inputs	• name – the Sort Algorithm name, as it has be registered in the system
Outputs	• Sort – an instance of the sort extension.
Post-Conditions	• The fully initialized Sort Algorithm is returned to the consumer
Exception Conditions	• The input 'name' does not pass validation as noted above

3.3.2.12 getSortAlgorithms

Returns a description of all registered extensions used to provide additional sorting of query results in the given context.

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns a list of Sort Algorithms that have been registered in the system • Validation is carried out to ensure that: o The context provided (if any) is a valid SortContext • If all conditions are met, the list of descriptions of the registered Sort Algorithms is returned to the user
Pre-Conditions	• Any Sort Algorithms must be available and registered in the system to be returned
Inputs	• context – the SortContext to restrict results to. If not provided, no SortContext restriction will be considered
Outputs	• SortDescriptionList - The list of descriptions of the registered Sort Algorithms in the system
Post-Conditions	• None
Exception Conditions	• The input 'context' does not pass validation as noted above

3.3.2.13 getSupportedCodingSchemes

Return a list of coding schemes and versions that are supported by this service, along with their status

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns a list of CodingSchemes that have been registered in the system
Pre-Conditions	• CodingSchemes available to be returned via this operation must be loaded and registered correctly in the system
Inputs	• None
Outputs	• CodingSchemeRenderingList - The list of CodingScheme descriptions and versions for every correctly loaded and registered CodingScheme in the system
Post-Conditions	• none
Exception Conditions	• The underlying system fails to initialize

3.3.2.14 resolveCodingScheme

Return detailed coding scheme information given a specific tag or version identifier.

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns the LexGrid:CodingScheme given the input parameters • Validation will ensure that: o The Requested Coding Scheme exists in the System o The specified version of the Coding Scheme is available o The specified Coding Scheme is Active
Pre-Conditions	• The requested CodingScheme: o Has been loaded and registered in the system o Is Active
Inputs	• codingScheme - The local name, URI, or formal name of the requested CodingScheme • versionOrTag - A LexGrid:CodingSchemeVersionOrTag indicating the Version or Tag of the Requested CodingScheme. o If not provided, and there exists only one CodingScheme in the system matching the

	‘codingScheme’ input, that CodingScheme will be used. • If more than one CodingScheme in the system matches the ‘codingScheme’ input, the CodingScheme tagged as “PRODUCTION” will be used.
Outputs	• The LexGrid:CodingScheme give the requested inputs
Post-Conditions	• A LexGrid:CodingScheme matching the requested parameters is returned
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above
Note	• Because of size concerns, the LexGrid:CodingScheme may or may not be fully populated, based on the underlying implementation. For example, all LexGrid:Entities may not be returned, as the size would be prohibitive for the intent of this operation

3.3.2.15 resolveCodingSchemeCopyright

Return coding scheme copyright given a specific tag or version identifier

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns the Copyright representation of the requested CodingScheme • Validation will ensure that: o The Requested Coding Scheme exists in the System o The specified version of the Coding Scheme is available o The specified Coding Scheme is Active
Pre-Conditions	• The requested CodingScheme: o Has been loaded and registered in the system o Is Active
Inputs	• codingScheme - The local name, URI, or formal name of the requested CodingScheme • versionOrTag - A LexGrid:CodingSchemeVersionOrTag indicating the Version or Tag of the Requested CodingScheme. o If not provided, and there exists only one CodingScheme in the system matching the ‘codingScheme’ input, that CodingScheme will be used. • If more than one CodingScheme in the system matches the ‘codingScheme’ input, the CodingScheme tagged as

	"PRODUCTION" will be used.
Outputs	• The LexGrid:CodingSchemeCopyRight – the copyright of the coding scheme given the requested inputs
Post-Conditions	• A representation of the Copyright of the requested CodingScheme is returned and available to the user
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above
Note	• If the implementation provides any security measures to restrict access to CodingSchemes (for example, restrict CodingScheme access based on the license of the CodingScheme), this operation should not be constrained by these security measures. The intent is to show Copyrights even for security protected CodingSchemes

3.3.2.16 setSecurityToken

Return coding scheme copyright given a specific tag or version identifier

Links to: JavaDoc and Schema XSD

Behavior Description	• Registers a security token for a coding scheme.
Pre-Conditions	• The requested CodingScheme: o Has been loaded and registered in the system o Is Active
Inputs	• codingScheme - The local name, URI, or formal name of the requested CodingScheme • token - The assigned security token.
Outputs	• LexBIGServiceGrid – an instance of LexBIGServiceGrid.
Post-Conditions	• An instance of the LexBIGServiceGrid of the requested CodingScheme is returned and available to the user.
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above
Note	•

3.3.3 Operations Details for CodedNodeGraphGrid

A virtual graph where the edges represent associations and the nodes represent coded entries. A CodedNodeGraph describes a graph that can be combined with other graphs, queried or resolved into an actual graph rendering

Links to: JavaDoc and UML

3.3.3.1 areCodesRelated

Determine whether there is a directed edge (or transitive closure of an edge) from the source code to the target code in this graph. The last parameter determines whether only direct associations are considered or whether the transitive closure of the edge is used.

Links to: JavaDoc and Schema XSD

Behavior Description	• Determine whether there is a directed edge (or transitive closure of an edge) from the source code to the target code in this graph. • The last parameter determines whether only direct associations are considered or whether the transitive closure of the edge is used.
Pre-Conditions	• The CodedNodeGraph has been initialized for the given CodingScheme
Inputs	• policy - Policy for resolving the relationship • association - Identifies the association to be tested. The name and value will be compared against the local name and URN of supported associations for participating coding schemes.
Outputs	• CodeRelationship
Post-Conditions	• none
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.3.2 intersect

Return the set of nodes and associations that are present in both graphs.

Links to: JavaDoc and Schema XSD

Behavior Description	• Adds a Intersect Restriction to the CodedNodeGraphGrid
Pre-Conditions	• Both the target CodedNodeGraphGrid and the source CodedNodeGraphGrid involved in the Intersection have been properly initialized.

Inputs	- graph Identifies the CodedNodeGraphGrid to be intersected with.
Outputs	- A new CodedNodeGraphGrid representing the intersection result.
Post-Conditions	- The resulting CodedNodeGraphGrid is properly initialized and able to be resolved. - Any restrictions placed on the CodedNodeGraphGrid involved in the restriction are preserved
Exception Conditions	- The underlying system fails to initialize

3.3.3.3 isCodeInGraph

Determine whether the supplied code is in the graph.

Links to: JavaDoc and Schema XSD

Behavior Description	- Returns whether or not, given the restrictions that have currently been place on the CodedNodeGraphGrid, the given node exists in the graph
Pre-Conditions	- The CodedNodeGraphGrid has been initialized, and all appropriated Restrictions have been placed on the graph prior to invoking this operation
Inputs	- code Identifies the coding scheme and code to test.
Outputs	- CodeExistance - True if the code is present; otherwise False.
Post-Conditions	- The original CodedNodeGraphGrid is unchanged, and maybe be Restricted further, or Resolved.
Exception Conditions	- The underlying system fails to initialize

3.3.3.4 listCodeRelationships

Return a list of all of the associations in the graph that have the supplied source and target codes or, if directOnly is false, all associations whose transitive closure has the supplied associations.

Links to: JavaDoc and Schema XSD

Behavior	- Computes node edges, as described above. - Note: all Restrictions placed on the graph (if any) will be

Description	respected and incorporated into node relationship calculations
Pre-Conditions	- The CodedNodeGraphGrid has been initialized, and all appropriated Restrictions have been placed on the graph prior to invoking this operation
Inputs	- policy - Policy for resolving the relationship
Outputs	- List - The list of code references for matching associations
Post-Conditions	- The original CodedNodeGraphGrid is unchanged, and maybe be Restricted further, or Resolved.
Exception Conditions	- The underlying system fails to initialize
Note	- Note that while the class of the returned value appears to imply concepts only, each contained reference inherits from the more general CodedNodeReference and is capable of representing any type of node contained by the graph.

3.3.3.5 listCodeRelationships

Return a list of all of the associations in the graph that have the supplied source and target codes based on distance between them. Distance (or the No. of edges) for a direct association between a source and target codes is 1. Values if distance should be equal or greater than 1, otherwise exception is thrown. Resulting list is not based on associations source & target have, but on distance only.

Links to: JavaDoc and Schema XSD

Behavior Description	- Computes node edges, as described above. - Note: all Restrictions placed on the graph (if any) will be respected and incorporated into node relationship calculations
Pre-Conditions	- The CodedNodeGraphGrid has been initialized, and all appropriated Restrictions have been placed on the graph prior to invoking this operation
Inputs	- policy - Policy for resolving the relationship
Outputs	- ConceptReferenceList - The list of code references for matching associations
Post-Conditions	- The original CodedNodeGraphGrid is unchanged, and maybe be Restricted further, or Resolved.
Exception Conditions	- The underlying system fails to initialize

Note	• Note that while the class of the returned value appears to imply concepts only, each contained reference inherits from the more general CodedNodeReference and is capable of representing any type of node contained by the graph.

3.3.3.6 resolveAsList

Resolve all of the coded nodes in the list, sorting by the supplied property (if any), resolving the supplied properties, resolving coded entries to the supplied depth and resolving associations to the supplied depth.

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns the underlying graph, given any placed Restrictions, as described above • Validation will ensure that: o The Requested CodingScheme exists in the system o The Requested CodingScheme is not Active within the system. o If no focus is provided, parameters ‘resolveForward’ OR ‘resolveBackward’ may be True, or neither may be True, but both may not be True o ‘propertyNames’ are valid for the CodingScheme o ‘sortOptions’ are registered and available as Sort Extensions
Pre-Conditions	• The CodedNodeGraphGrid has been initialized, and all appropriated Restrictions have been placed on the graph prior to invoking this operation
Inputs	• policy - Policy for resolving the relationship
Outputs	ResolvedConceptReferenceList - A list of code references, up to the maximum number specified in the policy. Note that in the event that a maximum number 'n' is specified and exactly 'n' items are returned, there is currently no flag or notification provided to indicate whether all available items were returned. Each entry will include basic information for the node along with an embedded object (e.g. concept) populated with requested properties.
Post-Conditions	• The original CodedNodeGraphGrid is unchanged, and maybe be Restricted further, or Resolved again.
Exception Conditions	• The underlying system fails to initialize

Note	• Note that while the class of the returned value appears to imply concepts only, each contained reference inherits from the more general CodedNodeReference and is capable of representing any type of node contained by the graph.

3.3.3.7 restrictToAssociations

Restrict the graph to the nodes that participate as a source or target of the named association and, if supplied, the named association qualifiers.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the graph, as described above • Validation will ensure that: o The associations provided are valid associations for the CodingScheme o The association qualifiers provided (if any) are valid association qualifiers for the CodingScheme • Returns a new CodedNodeGraphGrid representing the filtered result.
Pre-Conditions	• The CodedNodeGraph has been initialized
Inputs	• Association List of associations used to restrict the graph. The name and value for each item in the list will be compared against the local name and URN of supported associations for participating coding schemes. • associationQualifiers If supplied, restriction only applies to associations that are qualified by one or more of the supplied qualifiers. The name and value for each item in the list will be compared against the local name and URN of supported association qualifiers for participating coding schemes.
Outputs	• CodedNodeGraphGrid - representing the filtered result
Post-Conditions	• The CodedNodeGraphGrid may be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.3.8 restrictToDirectionalNames

Restrict the graph to the nodes that participate as a source or target of an association whose directional name matches the one provided and, if supplied, the named association qualifiers. A directional name is considered to be either the forward or reverse label registered to an association defined by the ontology. Forward and reverse names are optionally assigned to each association. For example, an association 'lineage' may have a forward name 'ancestorOf' and reverse name 'descendantOf'

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the graph, as described above • Validation will ensure that: o The associations provided are valid associations for the CodingScheme o The association qualifiers provided (if any) are valid association qualifiers for the CodingScheme • Returns a new CodedNodeGraphGrid representing the filtered result.
Pre-Conditions	• The CodedNodeGraphGrid has been initialized
Inputs	• directionalNames List of directionalNames used to restrict the graph. A directional name is compared against the forward and reverse names for defined associations. If a given name matches more than one forward or reverse label, all corresponding associations are included in the restriction. • associationQualifiers If supplied, restriction only applies to associations that are qualified by one or more of the supplied qualifiers. The name and value for each item in the list will be compared against the id and URN of supported association qualifiers for participating coding schemes.
Outputs	• CodedNodeGraphGrid representing the filtered result
Post-Conditions	• The CodedNodeGraphGrid may be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.3.9 restrictToCodes

Return a graph that contains only the codes that are present in the supplied list, and all edges that still have a source and target code remaining.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the graph, as described above • Returns a new CodedNodeGraphGrid representing the filtered result.
Pre-Conditions	• The CodedNodeGraph has been initialized
Inputs	• codes Codes to filter on.
Outputs	• CodedNodeGraphGrid representing the filtered result
Post-Conditions	• The CodedNodeGraphGrid may be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.3.10 restrictToCodeSystem

Restrict the graph to codes (source and target) that originate from the supplied code system. Note: edges defined by other code systems will still be resolved if associated with both source and target nodes for the restricted code system.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the graph, as described above • Returns a new CodedNodeGraphGrid representing the filtered result.
Pre-Conditions	• The CodedNodeGraphGrid has been initialized
Inputs	• codingScheme The local name or URN of the coding scheme to filter on.
Outputs	• CodedNodeGraphGrid representing the filtered result
Post-Conditions	• The CodedNodeGraphGrid may be Restricted further, or Resolved.

Exception Conditions	• The underlying system fails to initialize

3.3.3.11 restrictToSourceCodes

Restrict the graph to associations that have one of the codes in the supplied list as source codes.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the graph, as described above • Returns a new CodedNodeGraphGrid representing the filtered result.
Pre-Conditions	• The CodedNodeGraph has been initialized
Inputs	• codes Codes to filter on.
Outputs	• CodedNodeGraphGrid representing the filtered result
Post-Conditions	• The CodedNodeGraphGrid may be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize

3.3.3.12 restrictToSourceCodeSystem

Restrict the graph to edges that have codes from the specified code system as a source.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the graph, as described above • Returns a new CodedNodeGraphGrid representing the filtered result.
Pre-Conditions	• The CodedNodeGraph has been initialized
Inputs	• codingScheme The local name or URN of the coding scheme to filter on.
Outputs	• CodedNodeGraphGrid representing the filtered result

Post-Conditions	• The CodedNodeGraphGrid may be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize

3.3.3.13 restrictToTargetCodes

Restrict the graph to associations that have one of the codes in the supplied list as target codes.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the graph, as described above • Returns a new CodedNodeGraphGrid representing the filtered result.
Pre-Conditions	• The CodedNodeGraph has been initialized
Inputs	• codes Codes to filter on.
Outputs	• CodedNodeGraphGrid representing the filtered result
Post-Conditions	• The CodedNodeGraphGrid may be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize

3.3.3.14 restrictToTargetCodeSystem

Restrict the graph to edges that have codes from the specified code system as a target.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the graph, as described above • Returns a new CodedNodeGraphGrid representing the filtered result.
Pre-Conditions	• The CodedNodeGraphGrid has been initialized
Inputs	• codingScheme The local name or URN of the coding scheme to filter on.

Outputs	- CodedNodeGraphGrid representing the filtered result
Post-Conditions	- The CodedNodeGraphGrid may be Restricted further, or Resolved.
Exception Conditions	- The underlying system fails to initialize

3.3.3.15 toNodeList

Transform the graph into a simple of list of code references, removing all association information.

Links to: JavaDoc and Schema XSD

Behavior Description	- Returns the underlying graph as a list of nodes, given any placed Restrictions. - Validation will ensure that: o The Requested CodingScheme exists in the system o The Requested CodingScheme is not Active within the system. o If no focus is provided, parameters 'resolveForward' OR 'resolveBackward' may be True, or neither may be True, but both may not be True
Pre-Conditions	- The CodedNodeGraphGrid has been initialized, and all appropriated Restrictions have been placed on the graph prior to invoking this operation
Inputs	- policy - Policy for resolving the relationship
Outputs	CodedNodeSetGrid - A set with matching items, up to the maximum number specified in the policy. Note that in the event that a maximum number 'n' is specified and exactly 'n' items are returned, there is currently no flag or notification provided to indicate whether all available items were returned.
Post-Conditions	- The original CodedNodeGraphGrid is unchanged, and maybe be Restricted further, or Resolved again.
Exception Conditions	- The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.3.16 union

Return the union of the two graphs. Union, in this context, means that the resulting graph contains the unique set of coded entries (String independent) that are present in one or both of the graphs, and the unique combination of edges (associations) present in one or both of the graphs.

Links to: JavaDoc and Schema XSD

Behavior Description	- Adds a Union Restriction to the CodedNodeGraphGrid
Pre-Conditions	- Both the target CodedNodeGraphGrid and the source CodedNodeGraphGrid involved in the Union have been properly initialized.
Inputs	- graph Identifies the CodedNodeGraphGrid to be unioned.
Outputs	- CodedNodeGraphGrid representing the merged result.
Post-Conditions	- The resulting CodedNodeGraphGrid is properly initialized and able to be resolved. - Any restrictions placed on the CodedNodeGraphGrid involved in the restriction are preserved
Exception Conditions	- The underlying system fails to initialize

3.3.4 Operations Details for CodedNodeSetGrid

A coded node set represents a flat list of coded entries.

Links to: JavaDoc and UML

3.3.4.1 difference

Return a coded node set that represents the set of concepts in this coded node set that are not included by the given set of codes.

Links to: JavaDoc and Schema XSD

Behavior Description	- Returns the difference of two CodedNodeSetGrid, given the input criteria
Pre-Conditions	- Both CodedNodeSetGrid participating in the difference operation have been initialized for the given CodingScheme
Inputs	- codesToRemove – List of codes to remove from the surrounding set.

Outputs	• CodedNodeSetGrid representing the difference.
Post-Conditions	• The resulting CodedNodeSetGrid is properly initialized and able to be resolved. • Any restrictions placed on the CodedNodeSetGrid involved in the restriction are preserved
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.4.2 intersect

Return a coded node set that represents the set of concepts that this node set and the provided node set have in common.

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns the intersection of two CodedNodeSetGrid, given the input criteria
Pre-Conditions	• Both CodedNodeSetGrid participating in the intersection operation have been initialized for the given CodingScheme
Inputs	• codes – Set of codes to intersect
Outputs	• CodedNodeSetGrid representing the intersection result
Post-Conditions	• The resulting CodedNodeSetGrid is properly initialized and able to be resolved. • Any restrictions placed on the CodedNodeSetGrid involved in the restriction are preserved
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.4.3 isCodeInSet

Return true if the supplied concept reference is contained within the represented list.

Links to: JavaDoc and Schema XSD

Behavior	• Returns whether or not, given the restrictions that have

Description	currently been place on the CodedNodeSetGrid, the given node exists in the graph
Pre-Conditions	• The CodedNodeSetGrid has been initialized, and all appropriated Restrictions have been placed on the set prior to invoking this operation
Inputs	• Code Coding scheme and concept code to test
Outputs	• CodeExsitance - True if the code is present; otherwise False.
Post-Conditions	• The original CodedNodeSetGrid is unchanged, and maybe be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize

3.3.4.4 resolve

Resolve an iterator over concepts matching the given criteria.

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns the underlying node set, given any placed Restrictions, as described above • Validation will ensure that: o The Requested CodingScheme exists in the system o The Requested CodingScheme is not Active within the system. o ‘sortOptions’ are valid and registered in the system o ‘propertyNames’ are valid for the CodingScheme
Pre-Conditions	• The CodedNodeSetGrid has been initialized, and all appropriated Restrictions have been placed on the set prior to invoking this operation
Inputs	• policy - Policy for resolving the CodedNodeSet
Outputs	• ResolvedConceptReferencesIterator - An iterator over matching entries. Each entry will include basic information for the node along with an embedded object (e.g. concept) populated with requested properties.
Post-Conditions	• The original CodedNodeSetGrid is unchanged, and may be Restricted further, or Resolved again.
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above
Note	• Note that while the class of the returned value appears to imply concepts only, each contained reference inherits

	from the more general CodedNodeReference and is capable of representing any type of node contained by the set.

3.3.4.5 resolveToList

Resolve the set to a list of concepts sorted by the supplied parameters, resolving all of the properties named in the list.

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns the underlying node set, given any placed Restrictions, as described above • Validation will ensure that: o The Requested CodingScheme exists in the system o The Requested CodingScheme is not Active within the system. o 'sortOptions' are valid and registered in the system o 'propertyNames' are valid for the CodingScheme
Pre-Conditions	• The CodedNodeSetGrid has been initialized, and all appropriated Restrictions have been placed on the set prior to invoking this operation
Inputs	• policy - Policy for resolving the relationship
Outputs	• ResolvedConceptReferenceList - A list of node references, up to the maximum number specified. Note that in the event that a maximum number 'n' is specified and exactly 'n' items are resolved, there is currently no flag or notification provided to indicate the requested list is fully resolved.
Post-Conditions	• The original CodedNodeSetGrid is unchanged, and may be Restricted further, or Resolved again.
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above
Note	• Note that while the class of the returned value appears to imply concepts only, each contained reference inherits from the more general CodedNodeReference and is capable of representing any type of node contained by the set.

3.3.4.6 restrictToCodes

Restrict the set to the list of codes in the supplied reference list.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the node set. • Returns a new CodedNodeSetGrid representing the filtered result.
Pre-Conditions	• The CodedNodeSetGrid has been initialized
Inputs	• codeList The list of codes to filter on.
Outputs	• CodedNodeSetGrid representing the filtered result
Post-Conditions	• The CodedNodeSetGrid may be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.4.7 restrictToMatchingDesignations

Restrict the list to the set of concepts that have designations that match the supplied string, using the supplied matching algorithm and language.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the node set, as described above • Returns a new CodedNodeSetGrid representing the filtered result. • Validation will ensure that: o ‘matchAlgorithm’ is valid and registered in the system
Pre-Conditions	• The CodedNodeSetGrid has been initialized
Inputs	• matchText – Filter String - syntax is determined by the match algorithm • option – Indicates the designations to search (one of the enumerated type SearchDesignationOption). • matchAlgorithm – Local name of the match algorithm - possible algorithms are returned in

	LexBigService.getMatchAlgorithms(). - language – Language of search string. If missing, use the default language specified in the context.
Outputs	- CodedNodeSetGrid representing the filtered result
Post-Conditions	- The CodedNodeSetGrid may be Restricted further, or Resolved.
Exception Conditions	- The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.4.8 restrictToMatchingProperties

Remove all elements from the set that do not have one or more properties that match the given criteria. Note that while property name and type are often synchronized, the API allows for them to be differentiated. For concepts, there are 5 major types of properties that can be assigned ('Comments', 'Definitions', 'Instructions', 'Presentations', and 'Generic' properties which can represent vocabulary-specific name/value pairings). Often the name assigned to a property will match the property type (e.g. a Presentation named 'textualPresentation' or a Definition named 'definition'). However, names are not fixed (e.g. a Presentation property may be named 'text' or 'textualPresentation'). This method allows for query based on property name, type, or both. However, at least one name or type must be specified.

Links to: JavaDoc and Schema XSD

Behavior Description	- Places an association specific Restriction on the node set, as described above - Returns a new CodedNodeSetGrid representing the filtered result. - Validation will ensure that: o ‘propertyNames’ are valid and registered in the system o ‘matchAlgorithm’ is valid and registered in the system
Pre-Conditions	- The CodedNodeSetGrid has been initialized
Inputs	- propertyNames – Indicates the local names of properties to match. To be recognized, each provided name must be defined in the coding scheme metadata as part of the registered supported properties. If empty or null, all names are evaluated for the specified property

	types. Note that the meta-property 'conceptCode' can be specified in addition to specific named properties defined by the code system. If 'conceptCode' is specified, the matchAlgorithms 'exactMatch', 'contains' and 'luceneQuery' and 'RegExp' are allowed. Any other request results in 'luceneQuery' being used. • propertyTypes – Indicates whether to match specific property categories, regardless of the assigned name. Any of the enumerated PropertyType values can be specified. If empty or null, properties of all types are evaluated. • sourceList – Local names of sources to match; each must be defined in the supported sources for the coding scheme. Returned values must match at least one of the specified values. A null or empty value indicates to match against all available sources. • contextList – Local names of usage contexts to match; each must be defined in the supported contexts for the coding scheme. Returned values must match at least one of the specified values. A null or empty value indicates to match against all available contexts. • qualifierList – Name/value pairings of property qualifiers to match. Each name must be defined in the supported property qualifiers for the coding scheme. Returned values must match at least one of the name/value combinations. A null or empty value indicates to match against all property qualifiers. • matchText – Property text to match - syntax is determined by the algorithm. • matchAlgorithm – Local name of the match algorithm - possible algorithms are returned in LexBigService.getMatchAlgorithms(). • language – Language of search string. If missing, use the default language specified in the context.

Outputs	• CodedNodeSetGrid representing the filtered result
Post-Conditions	• The CodedNodeSetGrid may be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.4.9 restrictToProperties

Remove all elements from the set that don't have one or more properties that match the given criteria. Note that while property name and type are often synchronized, the API allows for them to be differentiated. For concepts, there are 5 major types of properties that can be assigned ('Comments', 'Definitions', 'Instructions', 'Presentations', and 'Generic' properties which can represent vocabulary-specific name/value pairings). Often the name assigned to a property will match the property type (e.g. a Presentation named 'textualPresentation' or a Definition named 'definition'). However, names are not fixed (e.g. a Presentation property may be named 'text' or 'textualPresentation'). This method allows for query based on property name, type, or both. However, at least one name or type must be specified.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the node set. • Returns a new CodedNodeSetGrid representing the filtered result. • Validation will ensure that: o ‘propertyNames’ are valid and registered in the system
Pre-Conditions	• The CodedNodeSet has been initialized
Inputs	• propertyList – Local names of properties to use in restriction; each must be defined in the supported properties for the coding scheme. • propertyTypes – Indicates whether to match specific property categories, regardless of the assigned name. Any of the enumerated PropertyType values can be specified. If empty or null, properties of all types are evaluated. • sourceList – Local names of sources to match; each must be defined in the supported sources for the coding scheme. Returned values must match at least one of the specified values. A null or empty value

	indicates to match against all available sources. • contextList – Local names of usage contexts to match; each must be defined in the supported contexts for the coding scheme. Returned values must match at least one of the specified values. A null or empty value indicates to match against all available contexts. • qualifierList – Name/value pairings of property qualifiers to match. Each name must be defined in the supported property qualifiers for the coding scheme. Returned values must match at least one of the name/value combinations. A null or empty value indicates to match against all property qualifiers.
Outputs	• CodedNodeSetGrid representing the filtered result
Post-Conditions	• The CodedNodeSetGrid may be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.4.10 restrictToStatus

Restrict the set to concepts matching the given status criteria.

Links to: JavaDoc and Schema XSD

Behavior Description	• Places an association specific Restriction on the node set, as described above • Returns a new CodedNodeSetGrid representing the filtered result.
Pre-Conditions	• The CodedNodeSetGrid has been initialized
Inputs	• `activeOption` – Indicates whether to include active codes, inactive codes, or both in the resolved result set (one of the enumerated type ActiveOption). This is matched against the 'isActive' field for CodedEntry instances in the code system. • `status` – Indicates zero or more status values to match. Provided values are compared using an exact match algorithm against the 'status' field for matching

	entities. If null or empty, the restriction is evaluated based only on the specified activeOption.
Outputs	• CodedNodeSetGrid representing the filtered result
Post-Conditions	• The CodedNodeSetGrid may be Restricted further, or Resolved.
Exception Conditions	• The underlying system fails to initialize, or the input parameters fail to validate as described above

3.3.4.11 union

Return the set union of all of the codes in the containing or the referenced set

Links to: JavaDoc and Schema XSD

Behavior Description	• Adds a Union Restriction to the CodedNodeSetGrid
Pre-Conditions	• Both the target CodedNodeSetGrid and the source CodedNodeSetGrid involved in the Union have been properly initialized.
Inputs	• codes Codes to add to the union
Outputs	• CodedNodeSetGrid representing the merged result.
Post-Conditions	• The resulting CodedNodeSetGrid is properly initialized and able to be resolved. • Any restrictions placed on the CodedNodeSetGrid involved in the restriction are preserved
Exception Conditions	• The underlying system fails to initialize

3.3.5 Operations Details for LexBIGServiceMetadataGrid

Interface to perform system-wide query over metadata for loaded code systems and providers.

Links to: JavaDoc and UML

3.3.5.1 listCodingSchemes

List the coding schemes that are represented in the metadata index.

Links to: JavaDoc and Schema XSD

Behavior Description	• Query the available service metadata, as described above
Pre-Conditions	• The LexBIGServiceMetadataGrid service has been initialized correctly
Inputs	• none
Outputs	• A LexBIG:AbsoluteCodingSchemeVersionReferenceList indicating the loaded service metadata
Post-Conditions	• none
Exception Conditions	• The LexBIGServiceMetadataGrid service, or underlying LexEVS service fail to initialize

3.3.5.2 restrictToCodingScheme

Restrict the search to a particular coding scheme.

Links to: JavaDoc and Schema XSD

Behavior Description	• Add a Restriction to the available service metadata, as described above
Pre-Conditions	• The LexBIGServiceMetadataGrid service has been initialized correctly
Inputs	• acsvr – The coding scheme to restrict the search to. You may provide the URN, the version, or both.
Outputs	• A new LexBIGServiceMetadataGrid representing the restricted result.
Post-Conditions	• The LexBIGServiceMetadataGrid may be Restricted further, or Resolved
Exception Conditions	• The LexBIGServiceMetadataGrid service, or underlying LexEVS service fail to initialize

3.3.5.3 restrictToProperties

Restrict the search to a particular property. Currently, this can be any element or attribute name from the OBO metadata schema

Links to: JavaDoc and Schema XSD

Behavior Description	- Add a Restriction to the available service metadata, as described above
Pre-Conditions	- The LexBIGServiceMetadataGrid service has been initialized correctly
Inputs	- properties – The set of properties to restrict the search to. If you provide multiple properties, it is treated as an OR search.
Outputs	- A new LexBIGServiceMetadataGrid representing the restricted result.
Post-Conditions	- The LexBIGServiceMetadataGrid may be Restricted further, or Resolved
Exception Conditions	- The LexBIGServiceMetadataGrid service, or underlying LexEVS service fail to initialize

3.3.5.4 restrictToPropertyParents

Restrict the search by the parents of the metadata elements. The OBO MetaData format is hierarchical - if you wish to restrict your search to properties that are under another property, provide the required property containers here.

Links to: JavaDoc and Schema XSD

Behavior Description	- Add a Restriction to the available service metadata, as described above
Pre-Conditions	- The LexBIGServiceMetadataGrid service has been initialized correctly
Inputs	- propertyParents – The containers to require as parents. For example, to restrict the search to "contacts" that are under "about" that is under "authority" - provide "authority" and "about". The order of the parents does not matter. Multiple parents are treated as an AND - so the result is required to be under each of the parents going up the parent tree.
Outputs	- A new LexBIGServiceMetadataGrid representing the restricted result.
Post-Conditions	- The LexBIGServiceMetadataGrid may be Restricted further, or Resolved
Exception	- The LexBIGServiceMetadataGrid service, or underlying LexEVS service fail to initialize

Conditions	

3.3.5.5 restrictToValue

Restrict the result to the metadata elements that match the supplied string, using the supplied matching algorithm

Links to: JavaDoc and Schema XSD

Behavior Description	• Add a Restriction to the available service metadata. • Validation will ensure that: o 'matchAlgorithm' is valid and registered in the system
Pre-Conditions	• The LexBIGServiceMetadataGrid service has been initialized correctly
Inputs	• matchText – The match text. Format is determined by the match algorithm. • matchAlgorithm – Local name of the match algorithm - possible algorithms are returned in LexBigService.getMatchAlgorithms().
Outputs	• A new LexBIGServiceMetadataGrid representing the restricted result.
Post-Conditions	• The LexBIGServiceMetadataGrid may be Restricted further, or Resolved
Exception Conditions	• The LexBIGServiceMetadataGrid service, or underlying LexEVS service fail to initialize, or the input parameters fail to validate as described above

3.3.5.6 resolve

Apply all of the restrictions, and return the result

Links to: JavaDoc and Schema XSD

Behavior Description	• Returns the underlying MetadataPropertyList, given any placed Restrictions, as described above
Pre-Conditions	• The LexBIGServiceMetadataGrid service has been initialized correctly
Inputs	• none

Outputs	- The resulting MetadataPropertyList
Post-Conditions	- The LexBIGServiceMetadata may be Restricted further, or Resolved again
Exception Conditions	- The LexBIGServiceMetadata service, or underlying LexEVS service fail to initialize

3.3.6 Operations Details for HistoryServiceGrid

The history service returns information about the change history of a coding scheme.

Links to: JavaDoc and UML

3.3.6.1 getAncestors

Return a list of baselines supported by this service that were released on or after the first supplied date and were released on or before the second date. Returned baselines are arranged in sequential order, from earliest to latest.

Links to: JavaDoc and Schema XSD

Behavior Description	- Query the available History service, as described above
Pre-Conditions	- The History service has been initialized correctly
Inputs	- `conceptReference`– The reference from which to begin the query.
Outputs	- A LexBIG:NCIChangeEventList containing the ancestors of the given ‘conceptReference’ parameter
Post-Conditions	- The HistoryService service state is maintained and may be queried again
Exception Conditions	- The History service, or underlying LexEVS service fail to initialize

3.3.6.2 getBaselines

Return a list of baselines supported by this service that were released on or after the first supplied date and were released on or before the second date. Returned baselines are arranged in sequential order, from earliest to latest.

Links to: JavaDoc and Schema XSD

Behavior Description	- Query the available History service, as described above
Pre-Conditions	- The History service has been initialized correctly
Inputs	- releasedAfter If present, only return baselines released on or after the supplied date. - releasedBefore If present, only return baselines that were released before the specified date
Outputs	- A LexBIG:SystemReleaseList containing the baselines of the given parameters
Post-Conditions	- The HistoryService service state is maintained and may be queried again
Exception Conditions	- The History service, or underlying LexEVS service fail to initialize

3.3.6.3 getConceptChangeVersions

Return a list of all of the coding scheme versions in which the supplied concept changed between the two supplied times (inclusive).

Links to: JavaDoc and Schema XSD

Behavior Description	- Query the available History service, as described above
Pre-Conditions	- The History service has been initialized correctly
Inputs	- conceptReference The concept to pull the versions out of - beginDate Begin date (inclusive) to check for version changes. If omitted, go to earliest recorded date - endDate Last date to check for changes in (inclusive). If omitted include all dates past and including

	beginDate
Outputs	• A CodingSchemeVersionList
Post-Conditions	• The HistoryService service state is maintained and may be queried again
Exception Conditions	• The History service, or underlying LexEVS service fail to initialize

3.3.6.4 getConceptCreationVersion

Return the coding scheme version in which the supplied concept was created.

Links to: JavaDoc and Schema XSD

Behavior Description	• Query the available History service, as described above
Pre-Conditions	• The History service has been initialized correctly
Inputs	• conceptReference The concept to pull the creation version of
Outputs	• A CodingSchemeVersion
Post-Conditions	• The HistoryService service state is maintained and may be queried again
Exception Conditions	• The History service, or underlying LexEVS service fail to initialize

3.3.6.5 getDescendants

Return the list of change events identifying the immediate descendants of the given concept reference.

Links to: JavaDoc and Schema XSD

Behavior Description	• Query the available History service, as described above
Pre-Conditions	• The History service has been initialized correctly
Inputs	• conceptReference The concept focus of the query

Outputs	• A NCIChangeEventList containing the immediate descendants of the 'conceptReference' parameter
Post-Conditions	• The HistoryService service state is maintained and may be queried again
Exception Conditions	• The History service, or underlying LexEVS service fail to initialize

3.3.6.6 getEarliestBaseline

Return the earliest baseline version in the list.

Links to: JavaDoc and Schema XSD

Behavior Description	• Query the available History service, as described above
Pre-Conditions	• The History service has been initialized correctly
Inputs	• none
Outputs	• A SystemRelease containing earliest baselines of the History service
Post-Conditions	• The HistoryService service state is maintained and may be queried again
Exception Conditions	• The History service, or underlying LexEVS service fail to initialize

3.3.6.7 getEditActionList

Return the list of available NCI-defined change events for the given concept and coding scheme version.

Links to: JavaDoc and Schema XSD

Behavior Description	• Query the available History service, as described above
Pre-Conditions	• The History service has been initialized correctly
Inputs	• conceptReference Optional concept to get the action list for. If omitted, all events for the given change set (represented by a coding scheme version) are returned.

	• codingSchemeVersion Version to get the action list for
Outputs	• An NCIChangeEventList containing the change events as described above.
Post-Conditions	• The HistoryService service state is maintained and may be queried again
Exception Conditions	• The History service, or underlying LexEVS service fail to initialize

3.3.6.8 getEditActionList

Return the list of available NCI-defined change events for the given concept and date range.

Links to: JavaDoc and Schema XSD

Behavior Description	• Query the available History service, as described above
Pre-Conditions	• The History service has been initialized correctly
Inputs	• conceptReference Optional concept to get the action list for. If omitted, all events for the given date range are returned. • beginDate Begin date (inclusive) to check for version changes. If omitted, go to earliest recorded date. • endDate Last date to check for changes in (inclusive). If omitted include all dates past and including beginDate.
Outputs	• An NCIChangeEventList containing the change events as described above.
Post-Conditions	• The HistoryService service state is maintained and may be queried again
Exception Conditions	• The History service, or underlying LexEVS service fail to initialize

3.3.6.9 getEditActionList

Return the list of NCI-defined change events for the given concept and release; empty if not applicable.

Links to: JavaDoc and Schema XSD

Behavior Description	• Query the available History service, as described above
Pre-Conditions	• The History service has been initialized correctly
Inputs	• conceptReference Optional concept to get the action list for. If omitted the actions for all registered concepts for the specified system release are returned. • releaseURN URN of the system release to retrieve the action list for.
Outputs	• An NCIChangeEventList containing the change events as described above.
Post-Conditions	• The HistoryService service state is maintained and may be queried again
Exception Conditions	• The History service, or underlying LexEVS service fail to initialize

3.3.6.10 getLatestBaseline

Get the latest baseline in the list.

Links to: JavaDoc and Schema XSD

Behavior Description	• Query the available History service, as described above
Pre-Conditions	• The History service has been initialized correctly
Inputs	• none
Outputs	• A SystemRelease containing latest baselines of the History service
Post-Conditions	• The HistoryService service state is maintained and may be queried again
Exception	• The History service, or underlying LexEVS service fail to

Conditions	initialize

3.3.6.11 getSystemRelease

Return detailed information about the particular system release.

Links to: JavaDoc and Schema XSD

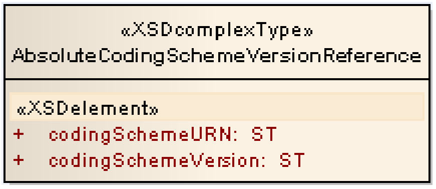
Behavior Description	- Query the available History service, as described above
Pre-Conditions	- The History service has been initialized correctly
Inputs	- releaseURN The URN of the system release to retrieve.
Outputs	- A SystemReleaseDetail of the Release URN specified by the 'releaseURN' parameter
Post-Conditions	- The HistoryService service state is maintained and may be queried again
Exception Conditions	- The History service, or underlying LexEVS service fail to initialize

3.4 Message Information Model

This section describes the information model of the messages sent to the LexEVS Analytical Grid service.

3.4.1 AbsoluteCodingSchemeVersionReference

AbsoluteCodingSchemeVersionReference represents an absolute reference to a coding scheme. This form of reference is service independent, as it doesn't depend on local coding schemes names or virtual tags.

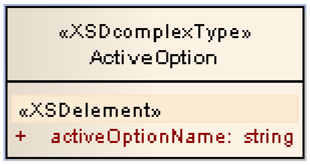

«XSDcomplexType»
AbsoluteCodingSchemeVersionReference
«XSDelement»
+ codingSchemeURN: ST
+ codingSchemeVersion: ST

Object Name	AbsoluteCodingSchemeVersionReference
Description of the Object	Represents an absolute reference to a coding scheme. This form of reference is service independent, as it doesn't depend on local

		coding schemes names or virtual tags.
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**
`codingSchemeURN`	ST	The URN of the referenced coding scheme.
`codingSchemeVersion`	ST	The Version of the referenced coding scheme.

3.4.2 ActiveOption

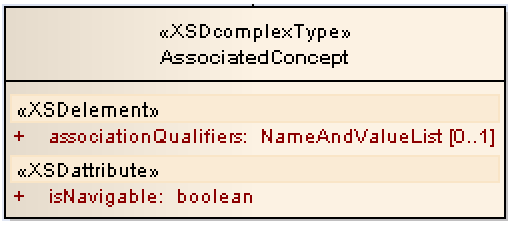
ActiveOption represents pre-determined options for filtering active, inactive or leaving filter neutral.



Object Name		ActiveOption
Description of the Object		Represents pre-determined options for filtering active, inactive or leaving filter neutral.
Link to the Object Specification		Enums.xsd
Attribute Name	**Type**	**Description**
`activeOptionName`	`String`	Options describing the status of a concept. Values must include: ACTIVE_ONLY, INACTIVE_ONLY, ALL

3.4.3 AssociatedConcept

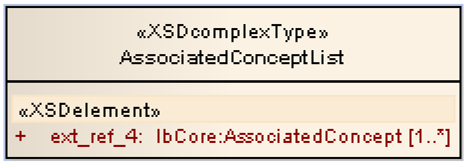
AssociatedConcept represents a concept reference that is the source or target of an association.



Object Name		AssociatedConcept
Description of the Object		Represents a concept reference that is the source or target of an association.
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**
`associationQualifiers`	`NameAndValueList`	Codes that qualify the complete association. Qualifiers may include "computed", degrees of certainty, etc.
`isNavigable`	`boolean`	True means that the association with this concept has been explicitly asserted and can be used for inferences. False means that the association was asserted in the other direction and cannot be used. Default: true

3.4.4 AssociatedConceptList

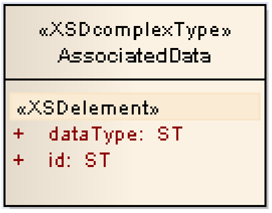
AssociatedConceptList represents a list of AssociatedConcept.



Object Name		AssociatedConceptList
Description of the Object		Represents a list AssociatedConcept
Link to the Object Specification		Collections.xsd
Attribute Name	**Type**	**Description**
`ext_ref_4`	`Array of AssociatedConcept`	List of AssociatedConcept

3.4.5 AssociatedData

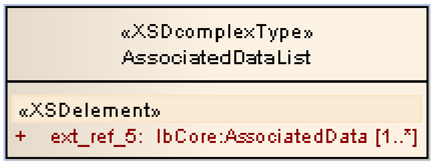
AssociatedData represents a "dataProperty" - data that is the target of an association.



Object Name		AssociatedData
Description of the Object		Represents a "dataProperty" - data that is the target of an association.
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**
dataType	ST	The data type of the property. The data itself is the value.
id	ST	An identifier that makes this chunk of data unique within an association.

3.4.6 AssociatedDataList

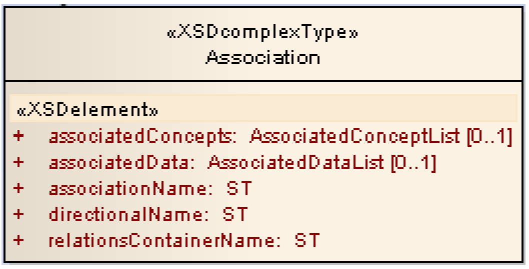
AssociatedDataList represents a list of AssociatedData.



Object Name		AssociatedDataList
Description of the Object		Represents a list AssociatedData
Link to the Object Specification		Collections.xsd
Attribute Name	**Type**	**Description**
ext_ref_5	Array of AssociatedData	List of AssociatedData

3.4.7 Association

Association represents a particular association as it appears in a CodedNode.

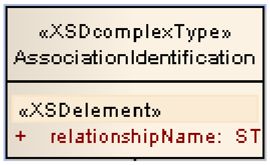

«XSDcomplexType»
Association
«XSDelement»
+ associatedConcepts: AssociatedConceptList [0..1]
+ associatedData: AssociatedDataList [0..1]
+ associationName: ST
+ directionalName: ST
+ relationsContainerName: ST

Object Name		Association
Description of the Object		Represents a particular association as it appears in a CodedNode
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**
associatedConcepts	AssociatedConceptList	The representation of The list of concepts that occur as the source or target of this association
associatedData	AssociatedDataList	The list of data elements that occur as the source or target of this association.
associationName	ST	The local name of the association
directionalName	ST	The name assigned to the association so that it can be read correctly going from the containing concept to the

		contained concept.
relationsContainerName	ST	The local name of the relations container

3.4.8 AssociationIdentification

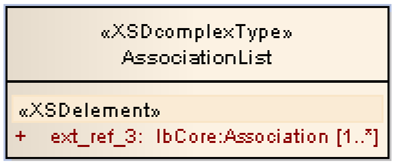
AssociationIdentification represents a unique identifier for an association within a coding scheme.



Object Name		AssociationIdentification
Description of the Object		Represents a unique identifier for an association within a coding scheme.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
relationshipName	ST	Name of a coding scheme relationship

3.4.9 AssociationList

AssociationList represents a list of Association.

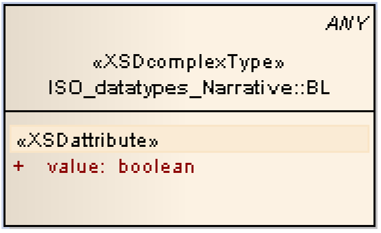


Object Name		AssociationList
Description of the Object		Represents a list Association
Link to the Object Specification		Collections.xsd
Attribute Name	**Type**	**Description**
ext_ref_3	Array of	List of Association

	Association	

3.4.10 BL

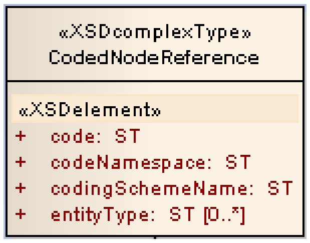
BL represents an ISO data type for Boolean value.



Object Name		BL
Description of the Object		Represents an ISO data type for Boolean value.
Link to the Object Specification		ISO_datatypes_Narrative.xsd
Attribute Name	**Type**	**Description**
value	boolean	Boolean value.

3.4.11 CodedNodeReference

CodedNodeReference represents a reference to an entity in the coding scheme by code, optionally qualified by namespace and type.

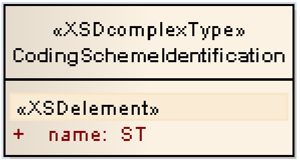


Object Name		CodedNodeReference
Description of the Object		Represents a reference to an entity in the coding scheme by code, optionally qualified by namespace and type.
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**

code	ST	The code uniquely identifying the entity within the coding scheme.
codeNamespace	ST	Local identifier of the code namespace. If omitted, namespace is implied from the URI of the coding scheme.
codingSchemeName	ST	The name of the coding scheme containing the entity
entityType	Array Of ST	Local identifiers of the types(s) defining the referenced entity (e.g. 'concept', 'instance').

3.4.12 CodingSchemeIdentification

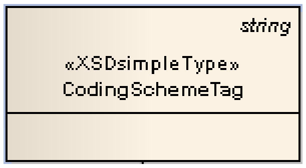
CodingSchemeIdentification represents an identity of a coding scheme.



Object Name		CodingSchemeIdentification
Description of the Object		Represents an identity of a coding scheme.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
name	ST	The name of a coding scheme. Usually descriptive or a mnemonic.

3.4.13 CodingSchemeTag

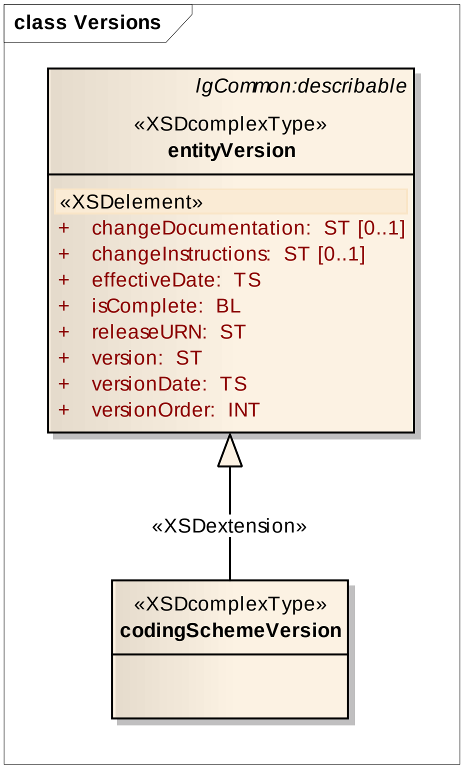
CodingSchemeTag represents user-defined tag that can symbolically identify the status of a given version of a coding scheme within a service. Typical values might include "staging", "development", "production", etc.



Object Name		CodingSchemeTag
Description of the Object		Represents a user-defined tag that can symbolically identify the status of a given version of a coding scheme within a service. Typical values might include "staging", "development", "production", etc.
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**

3.4.14 codingSchemeVersion

codingSchemeVersion represents a static snapshot of a coding scheme at a point in time.

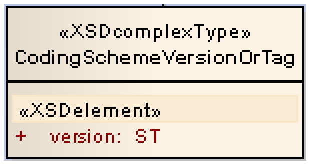


Object Name	codingSchemeVersion
Description of the Object	Represents a static snapshot of a coding scheme at a point in time.

Link to the Object Specification		Versions.xsd
Attribute Name	**Type**	**Description**
changeDocumentation	ST	User documentation about this particular change. Format is coding scheme specific.
changeInstructions	ST	Formal or semi-formal instructions on how to apply this change.
effectiveData	TS	The start date for which this version is operative (considered active).
isComplete	BL	If true, this entity represents a complete copy of the particular release. If false, it only carries a delta.
releaseURN	ST	URN of the release from which this version is drawn.
version	ST	The unique version identifier.
versionDate	TS	The end date for which this version is operative (considered committed).
versionOrder	INT	The relative order of this version within the surrounding container.

3.4.15 CodingSchemeVersionOrTag

CodingSchemeVersionOrTag represents a named coding scheme version or a virtual tag (e.g. latest, production, etc). Note that the tagged form of identifier is only applicable in the context of a given service, as one service may identify the scheme as "production" and another as "staging".



Object Name	CodingSchemeVersionOrTag
Description of the Object	Represents a named coding scheme version or a virtual tag (e.g. latest, production, etc). Note that the tagged form of identifier is only applicable in the

		context of a given service, as one service may identify the scheme as "production" and another as "staging".
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**
`version`	ST	The specific version of the coding scheme.
`tag`	ST	User-defined tag that can symbolically identify the status of a given version of a coding scheme within a service. Typical values might include "staging", "development", "production", etc.

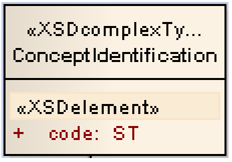
3.4.16 comment

comment is a property that is used as an annotation or other note about the state or usage of the entity. The propertyType of comment must be "comment"

Object Name		comment
Description of the Object		Represents a property that is used as an annotation or other note about the state or usage of the entity. The propertyType of comment must be "comment"
Link to the Object Specification		Concepts.xsd
Attribute Name	**Type**	**Description**

3.4.17 ConceptIdentification

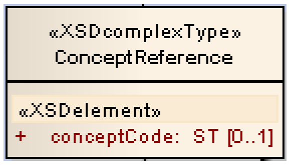
ConceptIdentification represents a unique identifier for a concept within a coding scheme.



Object Name		ConceptIdentification
Description of the Object		Represents a unique identifier for a concept within a coding scheme.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
code	ST	Text representation of a concept code

3.4.18 ConceptReference

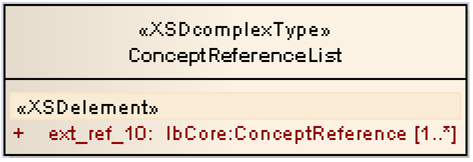
ConceptReference represents a reference to a concept code. Maintained for backward compatibility, but no longer enhanced in favor of the more flexible CodedNodeReference.



Object Name		ConceptReference
Description of the Object		Represents a reference to a concept code. Maintained for backward compatibility, but no longer enhanced in favor of the more flexible CodedNodeReference.
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**
conceptCode	ST	Uniquely identifies the concept within the coding scheme. Alias for CodedNodeReference:code.

3.4.19 ConceptReferenceList

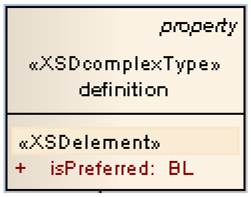
ConceptReferenceList represents a list of ConceptReference.



Object Name		ConceptReferenceList
Description of the Object		Represents a list ConceptReference
Link to the Object Specification		Collections.xsd
Attribute Name	**Type**	**Description**
`ext_ref_10`	`Array of ConceptReference`	List of ConceptReference

3.4.20 definition

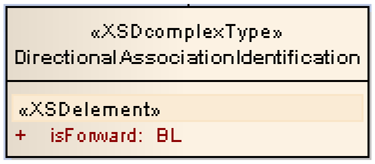
definition is a property that defines the entity in a particular language or context.. The propertyType of definition must be "definition"



Object Name		definition
Description of the Object		Represents a property that defines the entity in a particular language or context.. The propertyType of definition must be "definition"
Link to the Object Specification		Concepts.xsd
Attribute Name	**Type**	**Description**
`isPreferred`	`BL`	True means this is the preferred definition for the given language and context.

3.4.21 DirectionalAssociationIdentification

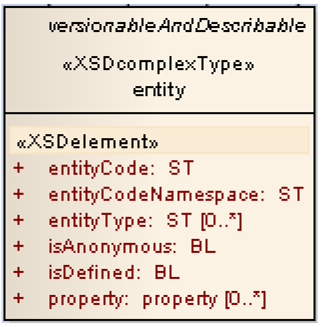
DirectionalAssociationIdentification represents a unique identifier for an association and the association direction within a coding scheme. DirectionalAssociationIdentification extends AssociationIdentification.



Object Name		DirectionalAssociationIdentification
Description of the Object		Represents a unique identifier for an association and association direction within a coding scheme.
Link to the Object Specification		caGrid.xsd
Attribute Name	Type	Description
isForward	BL	Direction Indication associated with an association name. If true this is a forward name. If false, this is a reverse name.

3.4.22 entity

entity represents a set of lexical assertions about the intended meaning of a particular entity code.

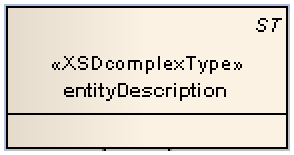


Object Name		entity
Description of the Object		Represents a set of lexical assertions about the intended meaning of a particular entity code.
Link to the Object Specification		Concepts.xsd
Attribute Name	Type	Description
entityCode	ST	The entity code being defined.

entityCodeNamespace	ST	Local identifier of the namespace of the entityCode. entityCodeNamespace must match a local id of a supportedNamespace in the corresponding mappings section. If omitted, the URI of the defaultCodingScheme will be used as the namespace for the entity code.
entityType	Array of ST	The local identifiers of the entity types(s) represented by this entity code. Must match a local id of a supportedEntityType in the corresponding mappings section.
isAnonymous	BL	True means that the entityCode is synthetic, and doesn't actually exist in the namespace. isAnonymous is used for synthetic top and bottom nodes as well as blank or anonymous inner class definitions. Default: False
isDefined	BL	True means that this entityCode is considered to be completely defined (i.e. necessary and sufficient) within the context of the containing code system. False means that only the necessary components are present. If omitted, the state of the entityCode definition is not known.
property	Array of property	The set of properties of this entityCode that are defined in the containing coding scheme.

3.4.23 entityDescription

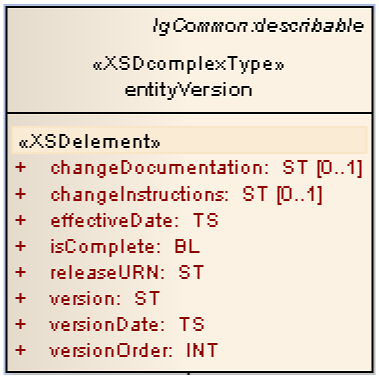
entityDescription represents the description of a resource. (Note: entityDescription may apply to any describable resource, not just "entities".



Object Name		entityDescription
Description of the Object		Represents the description of a resource.
Link to the Object Specification		CommonTypes.xsd
Attribute Name	**Type**	**Description**

3.4.24 entityVersion

entityVersion represents a static snapshot of the given entity.

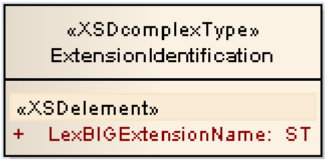


Object Name		entityVersion
Description of the Object		Represents a static snapshot of the given entity.
Link to the Object Specification		Versions.xsd
Attribute Name	**Type**	**Description**
changeDocumentation	ST	User documentation about this particular change. Format is coding scheme specific.
changeInstructions	ST	Formal or semi-formal instructions on how to apply this change.
effectiveData	TS	The start date for which this version is operative (considered active).
isComplete	BL	If true, this entity represents a complete copy of the particular release. If false, it

		only carries a delta.
releaseURN	ST	URN of the release from which this version is drawn.
version	ST	The unique version identifier.
versionDate	TS	The end date for which this version is operative (considered committed).
versionOrder	INT	The relative order of this version within the surrounding container.

3.4.25 ExtensionIdentification

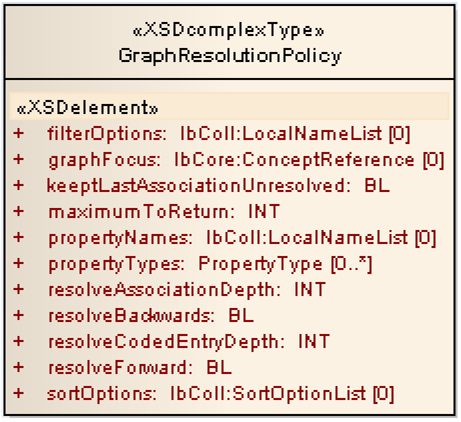
ExtensionIdentification represents an identity of a LexBIG extension. The LexBIG extension is a custom functionality added as an extension to LexEVS instance.



Object Name		ExtensionIdentification
Description of the Object		Represents an identity of a LexBIG extension. The LexBIG extension is a custom functionality added as an extension to LexEVS instance.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
LexBIGExtensionName	ST	Name of a class that extends functionality. This name serves as an identifier to the API to call through to the service layer for a particular extension class.

3.4.26 GraphResolutionPolicy

GraphResolutionPolicy represents the policy for resolving a node graph.


«XSDcomplexType»
GraphResolutionPolicy
«XSDelement»
+ filterOptions: lbColl:LocalNameList [0]
+ graphFocus: lbCore:ConceptReference [0]
+ keeptLastAssociationUnresolved: BL
+ maximumToReturn: INT
+ propertyNames: lbColl:LocalNameList [0]
+ propertyTypes: PropertyType [0..*]
+ resolveAssociationDepth: INT
+ resolveBackwards: BL
+ resolveCodedEntryDepth: INT
+ resolveForward: BL
+ sortOptions: lbColl:SortOptionList [0]

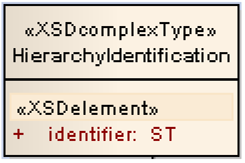
Object Name		GraphResolutionPolicy
Description of the Object		Represents the policy for resolving a node graph.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
filterOptions	LocalNameList	List of Filter extensions to apply during resolution. If supplied, filters are applied in the order provided. Each name in the list must correspond to the name of a Filter description as registered to the associated service. Available Filter descriptions can be retrieved through the LexBIGServiceGrid getFilterExtensions() method.
graphFocus	ConceptReference	Focus node of the graph. If present, only the nodes that are reachable via this node will be returned. If null, nodes with no incoming or outgoing associations are used as starting points for navigation of forward and reverse relationships, respectively.
keepLastAssociationUnresolved	BL	Keep the last hop while resolving associations to the resolveAssociationDepth unresolved. This is useful while drawing trees of an ontology and we need a quick way to tell if the tree can be expanded

		further.
maximumToReturn	INT	Maximum number of entries to return; a value less than 1 indicates to return unlimited entries (to the limit specified in the runtime configuration file).
propertyNames	LocalNameList	Indicates the local names of properties to match. To be recognized, each provided name must be defined in the coding scheme metadata as part of the registered supported properties. If empty or null, all names are evaluated for the specified property types.
propertyTypes	Array of PropertyType	Indicates whether to match specific property categories, regardless of the assigned name. Any of the enumerated PropertyType values can be specified. If empty or null, properties of all types are evaluated.
resolveAssociationDepth	INT	Number of hops to resolve associations.0 means leave all associations unresolved, 1 means immediate neighbors, etc. -1 means follow the entire closure of the graph.
resolveBackward	BL	True means resolve in the direction of target to source.
resolveCodedEntryDepth	INT	Number of hops to resolve associations. 0 means leave all associations unresolved, 1 means immediate neighbors, etc. -1 means follow the entire closure of the graph.
resolveForward	BL	True means resolve in the direction of source to target.
sortOptions	sortOptionList	List of sort options to apply during resolution. If supplied, the sort algorithms will be applied in the order provided. Any algorithms not valid to be applied in context of node set iteration, as specified in the sort

		extension description, will result in a parameter exception. Available algorithms can be retrieved through the LexBIGService getSortExtensions() method.

3.4.27 HierarchyIdentification

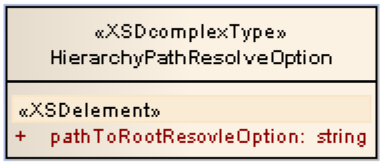
HierarchyIdentification represents a unique identifier for a coding scheme hierarchy.



Object Name		HierarchyIdentification
Description of the Object		Represents a unique identifier for a coding scheme hierarchy.
Link to the Object Specification		[caGrid.xsd]()
Attribute Name	**Type**	**Description**
`identifier`	`ST`	Identifier of a Hierarchy in a coding scheme.

3.4.28 HierarchyPathResolveOption

HierarchyPathResolveOption represents path to root resolve option.

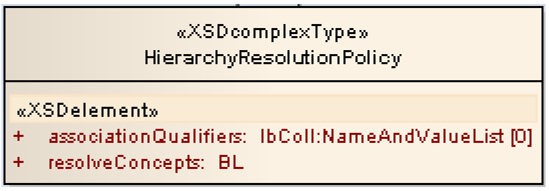


Object Name		HierarchyPathResolveOption
Description of the Object		Represents path to root resolve option.
Link to the Object Specification		[Enums.xsd]()
Attribute Name	**Type**	**Description**
`pathToRootResolveOption`	`String`	Resolve option for a hierarchy path to

		root. Use limited to the following options: ALL, ONE, ONE_PER_HIERARCHY, ONE_PER_ROOT

3.4.29 HierarchyResolutionPolicy

HierarchyResolutionPolicy represents a policy for resolving a list of associations from a given coding scheme hierarchy. It also links to ConceptIdentification and HierarchyIdentification.

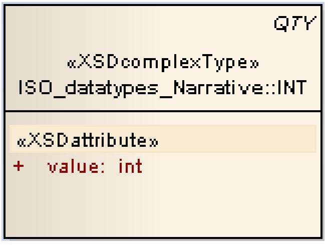

«XSDcomplexType»
HierarchyResolutionPolicy
«XSDelement»
+ associationQualifiers: lbColl:NameAndValueList [0]
+ resolveConcepts: BL

Object Name		HierarchyResolutionPolicy
Description of the Object		Represents a policy for resolving a list of associations from a given coding scheme hierarchy. It also links to ConceptIdentification and HierarchyIdentification
Link to the Object Specification		[caGrid.xsd]()
Attribute Name	**Type**	**Description**
`associationQualifiers`	`NameAndValueList`	List of association qualifiers to match.
`resolveConcepts`	`BL`	True to build and embed a full object (e.g. concept) for each referenced node in the returned results; false to return only basic identifying information (e.g. code, coding scheme, and description). If false, additional properties for referenced entries can be resolved on an item-by-item basis as controlled by the

		application.
hierarchyId	HierarchyIdentification	Identifier of a Hierarchy in a coding scheme.
conceptCode	ConceptIdentification	Identifier of a concept code

3.4.30 INT

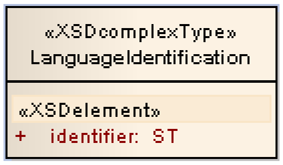
INT represents an ISO data type for integer.



Object Name		INT
Description of the Object		Represents an ISO data type for integer.
Link to the Object Specification		ISO_datatypes_Narrative.xsd
Attribute Name	**Type**	**Description**
value	int	Integer value.

3.4.31 LanguageIdentification

LanguageIdentification represents an identity of a language used in the terminology.

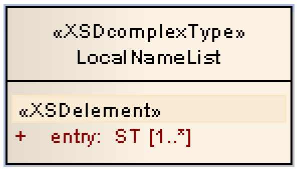


Object Name	LanguageIdentification
Description of the Object	Represents an identity of a language used in the terminology
Link to the Object	caGrid.xsd

Specification		
Attribute Name	**Type**	**Description**
identifier	ST	Identity of a language. Usually descriptive or a mnemonic.

3.4.32 LocalNameList

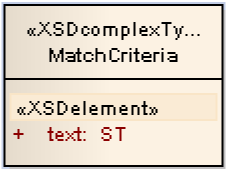
LocalNameList represents a list of local to coding scheme named entities.



Object Name		LocalNameList
Description of the Object		Represents a list of local to coding scheme named entities.
Link to the Object Specification		Collections.xsd
Attribute Name	**Type**	**Description**
entry	Array of ST	List of local names

3.4.33 MatchCriteria

MatchCriteria represents the value that needs to be matched.

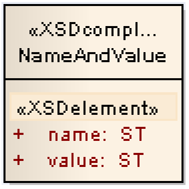


Object Name		MatchCriteria
Description of the Object		Represents the value that needs to be matched.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**

text	ST	String value that needs to be matched.

3.4.34 NameAndValue

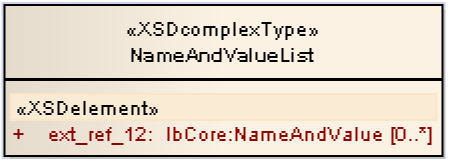
NameAndValue represents a simple name/value pair.



Object Name		NameAndValue
Description of the Object		Represents a simple name/value pair
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**
name	ST	A local name or other identifier
value	ST	Value attached to the name

3.4.35 NameAndValueList

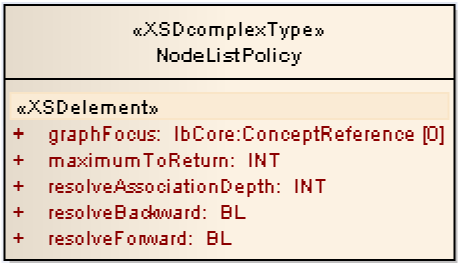
NameAndValueList represents a list of NameAndValue objects.



Object Name		NameAndValueList
Description of the Object		Represents a list of NameAndValue objects.
Link to the Object Specification		Collections.xsd
Attribute Name	**Type**	**Description**
ext_ref_12	Array of NameAndValue	List of NameAndValue

3.4.36 NodeListPolicy

NodeListPolicy represents the policy for resolving a node graph.

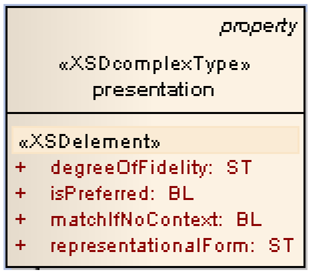


Object Name		NodeListPolicy
Description of the Object		Represents the policy for resolving a node graph.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
graphFocus	ConceptReference	Focus node of the graph. If present, only the nodes that are reachable via this node will be returned. If null, nodes with no incoming or outgoing associations are used as starting points for navigation of forward and reverse relationships, respectively.
maximumToReturn	INT	Maximum number of entries to return; a value less than 1 indicates to return unlimited entries (to the limit specified in the runtime configuration file).
resolveAssociationDepth	INT	Number of hops to resolve associations.0 means leave all associations unresolved, 1 means immediate neighbors, etc. -1 means follow the entire closure of the graph.

resolveBackward	BL	True means resolve in the direction of target to source.
resolveForward	BL	True means resolve in the direction of source to target.

3.4.37 presentation

presentation is a property that represents or designates the meaning of the entityCode. The propertyType of presentation must be "presentation"

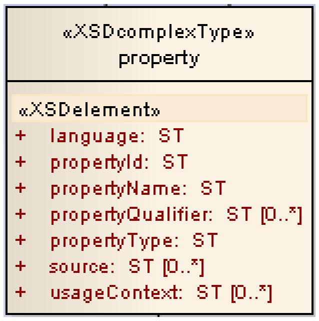


Object Name		presentation
Description of the Object		Represents or designates the meaning of the entityCode. The propertyType of presentation must be "presentation"
Link to the Object Specification		Concepts.xsd
Attribute Name	**Type**	**Description**
degreeOfFidelity	ST	A local identifier that states how closely a term approximates the intended meaning of an entry code. degreeOfFidelity must match a local id of a supportedDegreeOfFidelity in the corresponding mappings section.
isPreferred	BL	True means that, *if* the text meets the selection criteria, it should be the preferred form. For a given language there should be only one preferred presentation.
matchIfNoContext	BL	True means that this presentation is valid in a acontextual setting - that it is

		always valid in the given language. Default: true if there are no property usageContexts, false otherwise.
representationalForm	ST	A local identifier that states how the term represents the concept (abbrev, acronym, etc.) representationalForm must match a local id of a representationalForm in the corresponding mappings section.

3.4.38 property

property represents description, definition, annotation or other attribute that serves to further define or identify an resource.

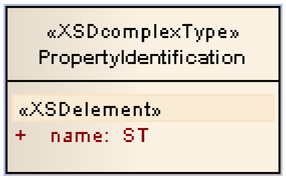


Object Name		property
Description of the Object		Represents description, definition, annotation or other attribute that serves to further define or identify an resource.
Link to the Object Specification		CommonTypes.xsd
Attribute Name	**Type**	**Description**
language	ST	The local identifier of the language of the property value. Must match a local id of a supportedLanguage in the corresponding mappings section. If omitted, and language is applicable to this property, the defaultLanguage of the surrounding resource is used.
propertyId	ST	A unique identifier of this particular propert/resource/value instance.

propertyName	ST	The local identifier that defines the meaning of this particular property entry. Must match a local id of a supportedProperty in the corresponding mappings section.
propertyQualifier	Array of ST	A qualifier that provides additional information about this particular property and/or its association with the resource.
propertyType	ST	The LexGrid model element that this property represents. As an example, the codingScheme "copyright" attribute could be represented by a property with a propertyType that mapped to lgCS:copyRight. Must match a local id of a supportedPropertyType in the corresponding mappings section.
source	Array of ST	The local identifiers of the source(s) of this property. Must match a local id of a supportedSource in the corresponding mappings section.
usageContext	Array of ST	The local identifiers of the context(s) in which this property applies. Must match a local id of a supportedContext in the corresponding mappings section.
text.dataType	ST	The local identifier of the format or data type of the text. Must match a local id of a supportedDataType in the corresponding mappings section. Default: tsCaseSensitiveIA5String
text.textValue	ST	The content

3.4.39 PropertyIdentification

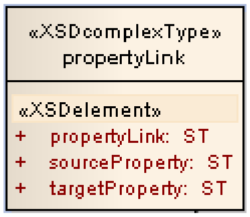
PropertyIdentification represents an identity of a property.



Object Name		PropertyIdentification
Description of the Object		Represents an identity of a property.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
name	ST	The name of a property. Usually descriptive or a mnemonic.

3.4.40 propertyLink

propertyLink represents a link between two properties for an entity. Examples include acronymFor, abbreviationOf, spellingVariantOf, etc. Must be in supportedPropertyLink.

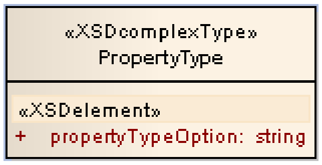


Object Name		propertyLink
Description of the Object		Represents a link between two properties for an entity. Examples include acronymFor, abbreviationOf, spellingVariantOf, etc. Must be in supportedPropertyLink.
Link to the Object Specification		Concepts.xsd
Attribute Name	**Type**	**Description**
propertyLink	ST	The local name of the type of linke between properties. propertyLink must match a local id of a supportedPropertyLink in the

		corresponding mappings section
sourceProperty	ST	The identifier of the first property in the link.
targetProperty	ST	The identifier of the second property in the link.

3.4.41 PropertyType

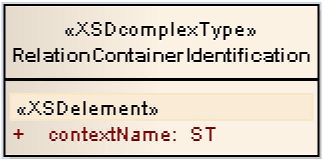
PropertyType represents options for filtering for pre-determined property types.



Object Name		PropertyType
Description of the Object		Represents options for filtering for pre-determined property types.
Link to the Object Specification		[Enums.xsd]()
Attribute Name	**Type**	**Description**
propertyTypeOption	String	Option from a set type of properties. Values can only be: COMMENT, DEFINITION, INSTRUCTION, PRESENTATION, GENERIC

3.4.42 RelationContainerIdentification

RelationContainerIdentification represents an identity of a relation container. The relation container contains set of associations between concepts.

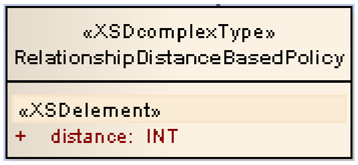


Object Name	RelationContainerIdentification
Description of the Object	Represents an identity of a relation container. The relation container contains set of associations

		between concepts.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
`contextName`	`ST`	The name of a relation container. Usually descriptive or a mnemonic.

3.4.43 RelationshipDistanceBasedPolicy

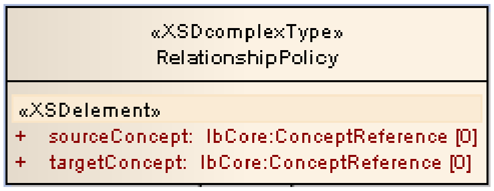
RelationshipDistanceBasedPolicy extends RelationshipPolicy and represents a distance between the source and target concepts.



Object Name		RelationshipDistanceBasedPolicy
Description of the Object		Extends RelationshipPolicy and represents a distance between the source and target concepts.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
`distance`	`INT`	Distance between the Source and Target concepts

3.4.44 RelationshipPolicy

RelationshipPolicy represents a policy for relationship between the source and target concepts.


«XSDcomplexType»
RelationshipPolicy
«XSDelement»
+ sourceConcept: lbCore:ConceptReference [0]
+ targetConcept: lbCore:ConceptReference [0]

Object Name	RelationshipPolicy

Description of the Object		Represents a policy for relationship between the source and target concepts.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
sourceConcept	ConceptReference	Source concept reference
targetConcept	ConceptReference	Target concept reference

3.4.45 RelationshipTypeBasedPolicy

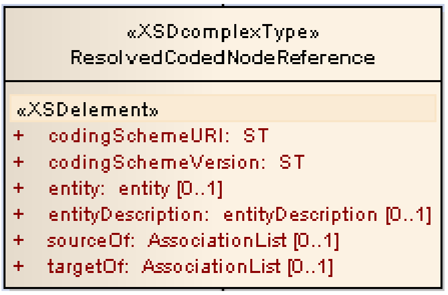
RelationshipTypeBasedPolicy extends RelationshipPolicy and represents the type of association between the source and target concepts.

«XSDcomplexType»
RelationshipTypeBasedPolicy
«XSDelement»
+ directOnly: BL

Object Name		RelationshipTypeBasedPolicy
Description of the Object		Extends RelationshipPolicy and represents the type of association between the source and target concepts.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
directOnly	BL	If True, the source and target concepts should be directly associated. If False, source and target concepts are transitively associated.

3.4.46 ResolvedCodedNodeReference

ResolvedCodedNodeReference represents a resolved information for an entity identified by coding scheme and code.

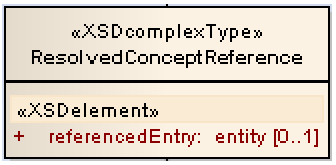

«XSDcomplexType»
ResolvedCodedNodeReference
«XSDelement»
+ codingSchemeURI: ST
+ codingSchemeVersion: ST
+ entity: entity [0..1]
+ entityDescription: entityDescription [0..1]
+ sourceOf: AssociationList [0..1]
+ targetOf: AssociationList [0..1]

Object Name		ResolvedCodedNodeReference
Description of the Object		Represents a resolved information for an entity identified by coding scheme and code.
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**
`codingSchemeURI`	`ST`	The URI of the associated coding scheme, if known.
`codingSchemeVersion`	`ST`	The version of the associated coding scheme, if known.
`entity`	`entity`	The referenced entity, optionally resolved.
`entityDescription`	`entityDescription`	Resolved information for an entity identified by coding scheme and code.
`sourceOf`	`AssociationList`	The list of associations for which the referenced code acts as the source (lhs, parent, ..) for. If this element is absent, no information is available about the target nodes. If the element is present, but 0 length, the code does not appear as the source of any associations (in the context of the containing graph).
`targetOf`	`AssociationList`	The list of associations for which the referenced code acts as the target (rhs, child..) for.

		If this element is absent, no information is available about the source nodes. If the element is present, but 0 length, the CodedNode doesn't appear as the target of any associations (in the context of the containing graph).

3.4.47 ResolvedConceptReference

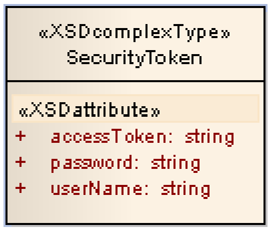
ResolvedConceptReference represents resolved information for a concept. Maintained for backward compatibility, but no longer enhanced in favor of the more flexible ResolvedCodedNodeReference.



Object Name		ResolvedConceptReference
Description of the Object		Represents resolved information for a concept. Maintained for backward compatibility, but no longer enhanced in favor of the more flexible ResolvedCodedNodeReference.
Link to the Object Specification		Core.xsd
Attribute Name	**Type**	**Description**
`referencedEntry`	`entity`	The resolved concept, if present. Alias for ResolvedCodedNodeReference:entity.

3.4.48 SecurityToken

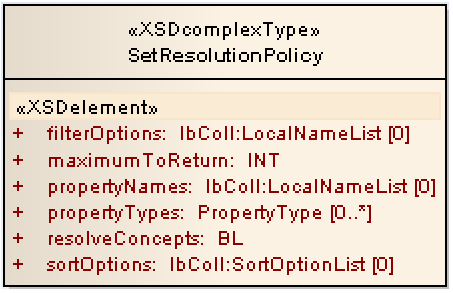
SecurityToken represents a security information needed to access secured terminology data.



Object Name		SecurityToken
Description of the Object		Represents a security information needed to access secured terminology data..
Link to the Object Specification		gov.nih.nci.evs.security.xsd
Attribute Name	**Type**	**Description**
accessToken	String	Access Token
password	String	Password
username	String	User name

3.4.49 SetResolutionPolicy

SetResolutionPolicy represents a policy to resolve a set of entries.

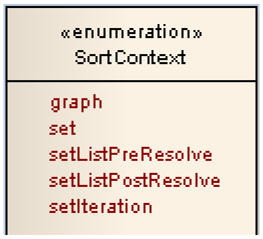

«XSDcomplexType»
SetResolutionPolicy
«XSDelement»
+ filterOptions: lbColl:LocalNameList [0]
+ maximumToReturn: INT
+ propertyNames: lbColl:LocalNameList [0]
+ propertyTypes: PropertyType [0..*]
+ resolveConcepts: BL
+ sortOptions: lbColl:SortOptionList [0]

Object Name		ST
Description of the Object		Represents a policy to resolve a set of entries.
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
filterOptions	LocalNameList	List of Filter extensions to apply during resolution. If supplied, filters are applied in the order provided. Each name in the list must correspond to the name of a Filter description as registered to the associated service. Available Filter descriptions can be retrieved through the LexBIGServiceGrid

		getFilterExtensions() method.
maximumToReturn	INT	Maximum number of entries to return; a value less than 1 indicates to return unlimited entries (to the limit specified in the runtime configuration file).
propertyNames	LocalNameList	Indicates the local names of properties to match. To be recognized, each provided name must be defined in the coding scheme metadata as part of the registered supported properties. If empty or null, all names are evaluated for the specified property types.
propertyTypes	Array of PropertyType	Indicates whether to match specific property categories, regardless of the assigned name. Any of the enumerated PropertyType values can be specified. If empty or null, properties of all types are evaluated.
resolveConcepts	BL	True to build and embed a full object (e.g. concept) for each referenced node in the returned results; false to return only basic identifying information (e.g. code, coding scheme, and description). If false, additional properties for referenced entries can be resolved on an item-by-item basis as controlled by the application.
sortOptions	SortOptionList	List of sort options to apply during resolution. If supplied, the sort algorithms will be applied in the order provided. Any algorithms not valid to be applied in context of node set iteration, as specified in the sort extension description, will result in a parameter exception. Available algorithms can be retrieved through the LexBIGService getSortExtensions() method.

3.4.50 SortContext

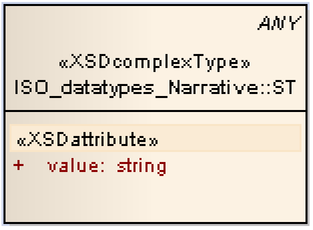
SortContext is an enumeration and represents a LexBIG sort module.



Object Name		SortContext
Description of the Object		SortContext is an enumeration and represents a LexBIG sort module.
Link to the Object Specification		[InterfaceElements.xsd]()
Attribute Name	**Type**	**Description**
`graph`	`String`	This sort is valid to be applied to resolved node graph representations.
`Set`	`String`	This sort is valid to be applied when resolving standard node set representations.
`setListPreResolve`	`String`	This sort is valid to be applied before the node set has been resolved to a list. All sorting is done BEFORE the results are resolved and returned, and takes into account ALL matching results, including those not included in the returned List.
`setListPostResolve`	`String`	This sort is valid to be applied after the node set has been resolved to a list. All sorting is done AFTER the results are resolved and returned, and takes into account ONLY those matching results in the List to be returned.
`setIteration`	`String`	This sort is valid to be applied when resolving node sets using iterators.

3.4.51 ST

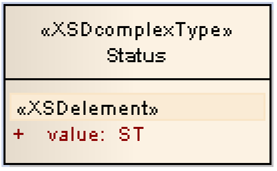
ST represents an ISO data type for String.



Object Name		ST
Description of the Object		Represents an ISO data type for String.
Link to the Object Specification		ISO_datatypes_Narrative.xsd
Attribute Name	**Type**	**Description**
value	String	String value.

3.4.52 Status

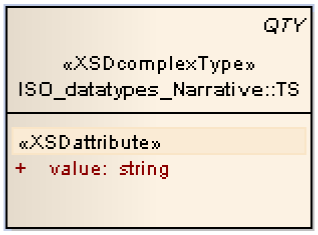
Status represents the current status code associated with a particular resource.



Object Name		Status
Description of the Object		Represents the current status code associated with a particular resource
Link to the Object Specification		caGrid.xsd
Attribute Name	**Type**	**Description**
value	ST	String value of a resource status.

3.4.53 TS

TS represents an ISO data type for Date.



Object Name		TS
Description of the Object		Represents an ISO data type for Date
Link to the Object Specification		ISO_datatypes_Narrative.xsd
Attribute Name	**Type**	**Description**
value	String	Date value.

3.5 Service Interactions

3.5.1 Actors

Name	Type	Description
LexEVS 6.0 API	System	LexEVS 6.0 Java RMI Vocabulary Content Services
LexEVS 6.0 Analytical Grid Services	System	LexEVS 6.0 Analytical Grid Services

3.5.2 Interaction Details

Actor Name	Producer / Consumer	Data	Link	Description
LexEVS 6.0 Analytical Grid Services	Consumer	Ontology Information.	Link to: JavaDoc	LexEVS 6.0 Analytical Grid Service

LexEVS 6.0 API	Producer	Ontology Information.	Link to: JavaDoc	The LexEVS Java RMI API provides the vocabulary content for the LexEVS Analytical Grid Service

3.6 Implementation Considerations

3.6.1 Security

Access Control	
Does the Service require Access Control mechanism to be in place to restrict access to only authenticated users or systems?	No
If Yes then provide the following in detail:	
N/A	
Application (Service) Security [Access Policy]	
Does the Service incorporate any security controls (Authorization) to ensure that access to information is granted to only the authorized users / systems?	Yes
If Yes then provide the following in detail:	
Certain vocabulary content accessible through the LexEVS Grid Service may require extra authorization to access. Each client is required to supply its own access credentials via Security Tokens. These Security Tokens are implemented by a SecurityToken object: Name: SecurityToken Namespace: gme://caCORE.caCORE/3.2/gov.nih.nci.evs.security Package: gov.nih.nci.evs.security Each call to "setSecurityToken" sets up a secured connection to Distributed LexEVS with the access privileges included in the SecurityToken parameter. The LexEVSGridServiceReference that is returned to the client contains a unique key	

identifier to the secure connection that has been created on the server. All subsequent calls the client makes through this LexEVSGridServiceReference will be made securely. If additional SecurityTokens are passed in through the “setSecurityToken” Grid Service, the additional security will be added and maintained. The “setSecurityToken” Grid Service is a stateful service. This means that after the client sets a SecurityToken, any subsequent call will be applied to that SecurityToken. Secure connections are not maintained on the server indefinitely, but are based on load conditions. The server will allow 30 unique secure connections to be set up for clients without any time limitations. As additional requests for secure connections are received by the server, connections will be released by the server on an ‘oldest first’ basis. No connection, however, may be released prior to 5 minutes after its creation. If no SecurityTokens are passed in by the client, a non-secure Distributed LexEVS connection will be used. The server maintains one (and only one) un-secured Distributed LexEVS connection that is shared by any client not requesting security. NOTE: All non-secured information accessed by the LexEVS Grid Service is publicly available from NCICB and users are expected to follow the licensing requirements currently in place for accessing and using NCI EVS information.	
Cryptography	
Does the Service require encryption of data transmitted to and from it?	No
If Yes then provide the following in detail:	
N/A	
Information Security and Risk Management	
Is the information served by the service confidential or privileged? And if yes, is it at risk from any external threats or vulnerabilities?	No
If Yes then provide the following in detail:	

N/A	
Legal, Regulations, Compliance and Investigations	
Does the information served by the service fall under any legal / regulatory compliance either at federal, state, local or institutional level ?	Yes
If Yes then provide the following in detail:	
Users are expected to follow the licensing requirements currently in place for accessing and using NCI EVS information.	
Telecommunications and Network Security	
Does the service need any network or transport level security such as SSL, Firewall protection etc.	Yes
If Yes then provide the following in detail:	
Firewall configurations required to provide access to the LexEVS 6.0 Analytical Grid Services.	

3.6.2 Auditing

No auditing requirements exist for LexEVS 6.0 Analytical Grid Services.

Operation Name	**Auditing Details**
N/A	•

Entity Name	Auditing Details
N/A	•

3.6.3 Privacy

No privacy requirements exist for LexEVS 6.0 Analytical Grid Services.

Data Element	Privacy Regulation	Security Control in Place	Access Requirement
N/A			

3.6.4 Error Handling

3.6.4.1 Overview

The LexEVS 6.0 Analytical Grid Service will conform to WSRF Error handling policies. For more information, see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf

3.6.4.2 Error Object Details

Error Object Name	LBInvocationException
Description of the Error Object	The exception to throw when invocation of a LexEVS service fails due to an unexpected problem captured and logged for administrative action. The logID will contain information that the LexEVS admins can use to track down the details of the internal error.
Link to the Object Specification	http://www.globus.org/toolkit/docs/4.0/execution/wsgram/schemas/ws_base_faults.html#type_BaseFaultType

Error Object	LBException

Name	
Description of the Error Object	Represents a LexEVS specific Exception occurring when the service can either: 1) Not complete the request because of missing client-supplied information. 2) Not complete the request because the requested content is unavailable This is a General LexEVS error, and is a Superclass for checked exceptions declared and thrown by the LexEVS runtime.
Link to the Object Specification	http://www.globus.org/toolkit/docs/4.0/execution/wsgram/schemas/ws_base_faults.html#type_BaseFaultType

Error Object Name	InvalidServiceContextAccess
Description of the Error Object	Represents the Exception occurring when a ServiceContext is accessed without being first initialized by the client.
Link to the Object Specification	http://www.globus.org/toolkit/docs/4.0/execution/wsgram/schemas/ws_base_faults.html#type_BaseFaultType

Error Object Name	LBParameterException
Description of the Error	Represents the Exception occurring when the client supplies either invalid or missing information as a parameter to a service function.

Object	
Link to the Object Specification	http://www.globus.org/toolkit/docs/4.0/execution/wsgram/schemas/ws_base_faults.html#type_BaseFaultType

Error Object Name	LBResourceUnavailableException
Description of the Error Object	Thrown when a resource required by the requested LexEVS operation cannot be located or resolved.
Link to the Object Specification	http://www.globus.org/toolkit/docs/4.0/execution/wsgram/schemas/ws_base_faults.html#type_BaseFaultType

4 Recommendations for Conformance and Compliance

4.1 Conformance Assertions

The following conformance assertions are required to conform to this LexEVS 6.0 Analytical Grid Services PSM.

No.	Name	Description	Test method
CS1	Conformance Profiles	In order to claim a minimal level of conformance to the LexEVS 6.0 Analytical Grid Service specification, designers/implementers are obligated to support the following mandatory Conformance Profile: QS-CP2 (LexEVS 21090 Full Query Conformance Profile)	1. Test cases to be defined to test each specified profile
CS2	ISO	Designers/implementers are	1. Test cases to include processing

No.	Name	Description	Test method
	Datatypes	obliged to support ISO 21090 datatypes in all operations	of each relevant data type

5 Appendix A - Relevant Standards

Standards	Description	Location
HL7 CTS 2	HL7's CTS 2 specification specifies functional model (CIM) outlining HL7's consensus requirement for terminology services. For the LexEVS CIM, PIM, and PSM, only the terminology and association query components of HL7 CTS 2 is considered to be in scope. LexEVS will ultimately implement much of the CTS 2 functionality, and as such, early identification of potential points of alignment is necessary.	Health Level Seven (HL7) Common Terminology Services – Release 2 (CTS 2)
ISO 21090 Health Informatics – Harmonized data types for information interchange	ISO 21090 data types provide a harmonized set of data type definitions for representing and exchanging healthcare related information. LexEVS 6.0 will interchange information using the 21090 data type specifications	http://www.kith.no/upload/4414/ISODIS21090.pdf

6 Appendix B - Glossary

Term	**Description**
Association	A binary relation from a set of entities to a set of entities and/or data.
Coding Scheme	A resource that makes assertions about a collection of terminological entities.
Property	A description, definition, annotation or other attribute that serves to further define or identify an resource.
RRF	UMLS Metathesaurus – Rich Release Format (RRF) (http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=nlmumls&part=ch03)
OWL	Web Ontology Language
OBO	The OBO flat file format is an ontology representation language.
Relation Identifier	A unique identifier of a relationship.
Registered Extensions	Used to implement application-specific behavior that is centrally accessible from a LexEVS service.
Service Reference (History, Coded Node Set, Coded Node Graph, Service Metadata)	A Service Reference is an abstract reference to a grouping of like functionality. A Service Reference will define intended behavior and capabilities of the Service, as well and provide a entry point for execution. Examples of a Service Reference would be a Java Interface, Web Service Endpoint Reference, an RMI (Remote Method Invocation) Endpoint, or any other abstract functional endpoint.
WSRF	Web Services Resource Framework (http://www.globus.org/wsrf).