

LexEVS 4.2 Release Notes

Contents of this Page

- [Release History](#)
- [New Features and Updates](#)
- [Bugs Fixed Since Last Release](#)
- [Known Issues](#)
- [Deferred/Deprecated Items](#)
- [CORE Product Dependencies](#)

**National Cancer Institute Center for
Bioinformatics
LexEVS 4.2 Release Notes
November 05 2008**

Release History

Release	Date
LexEVS 4.2	05 November 2008
EVS API 4.1.1	08 August 2008
EVS API 4.1	27 June 2008
EVS API 4.0	07 November 2007
caCORE 3.2	22 December 2006
caCORE 3.1	27 March 2006
caCORE 3.0.1.3	12 December 2006
caCORE 3.0.1.2	18 October 2005
caCORE 3.0.1.1	30 August 2005
caCORE 3.0.1	22 July 2005
caCORE 3.0	31 March 2005
caCORE 2.1	28 May 2004
caCORE 2.0.1	19 December 2003
caCORE 2.0	31 October 2003
caCORE 1.2	13 June 2003
caCORE 1.1	07 February 2003
caCORE 1.0	29 August 2002

New Features and Updates

DTSRPC V 2.2

Continues to provide legacy support for the caCORE 3.x EVS API.

NCI Thesaurus and MetaThesaurus Browsers

Continues to provide legacy support for the caCORE 3.2 EVS API.

LexEVS

The 4.2 release of LexEVS includes EVS API (based on the EVS 3.2 Object Model) with LexBIG v2.3.0 on the backend). The EVS API suite includes the caCORE SDK generated interfaces as well as, the Distributed LexBIG (DLB) API. The 4.2 release also includes the LexEVS 4.2 Grid Service which uses the DLB API to expose the underlying LexBIG interfaces.

[EVS:GF #12357](#) - Loader should allow Presentation properties to be specified

The LexBIG loader currently has a hard-coded list of property names that will be tagged as "Presentation" properties within LexBIG. Instead, the list of appropriate presentation properties should be set by the manifest.

[EVS:GF #8672](#) - Allow external configuration of complex properties

Per GForge #8671, additional support is to be added for complex properties (embedded as XML fragments in OWL source as part of the NCI curation process). The initial solution will involve customizing the NCI OWL loader to include enough knowledge to support the sources in caCORE 4.0.

[EVS:GF #13193](#) - Manifest support on additional loads (RRF, etc) or decoupled from load.

Manifest support on additional loads (RRF, etc) or decoupled from load.

Features:

- independent of loader
- tuning of code system metadata
- could be applied during and after the load

[EVS:GF #8442](#) - Lazy loading of properties for CodedEntry

Coding scheme data such as URN and Version are available in ResolvedConceptReference, but not in CodedEntry. This may present difficulties in implementing lazy loading of CodedEntry in the future. It is desirable to find the corresponding CodingScheme from CodedEntry itself.

[EVS:GF #13026](#) - Load both SAB and VSAB data for Meta sources

LexBIG only loads the SAB presentation of a source when loading data from Meta. We would like to have the VSAB information loaded as well. The end users want to know what version of the source that they are pulling data from.

[EVS:GF #7891](#) - Eliminate the dependencies of restriction methods (for example, CodedNodeSet restrictToMatchingProperties) on backend database

To improve the performance of distributed LexBIG, it is desirable that implementation of all restriction methods, such as restrictToMatchingProperties in CodedNodeSet, do not involve database access. Validation of input parameters, in this case, property names, may be performed at the resolve stage.

[EVS:GF #12995](#) - Add database info to GUI

Incorporate a way to view the dbName in the GUI. Either a column in the Available Coding Schemes window or a field in the Code System Viewer. This information is available in the registry.xml and the metadata.xml

[EVS:GF#13500](#) - MedDRA Loader

EVS will depend on the UMLS to obtain the version of MedDRA that we serve as a standalone terminology on LexBIG. A loader for the MedDRA terminology distribution needs to be developed (if necessary).

[EVS:GF#13501](#) - HL7 Loader

A LexBIG loader is needed for the HL7 XML format that is used to distribute version 2 and version 3 of the HL7 syntax terminology.

[EVS:GF#13565](#) - LOINC Loader

Extend load capability of the existing UMLS Loader to support loading of the LOINC terminology.

[EVS:GF#15071](#) - Make copyright statements accessible through LexBIGService, or ConvenienceMethods interface.

Currently, copyright or license statements are accessible through the getCopyright() method of CodingScheme. As a part of the GForge 10884 effort to safeguard licensed ontologies such as MedDRA from unauthorized access, the CodingScheme object currently accessible through the resolveCodingScheme method of LexBIGService would no longer be accessible without a valid authentication token. But, users would need to review copyright statements to understand any restriction on a licensed ontology. Therefore, it is necessary to modify LexBIG code base to make these statements accessible through the LexBIGService, or the ConvenienceMethods interface instead.

[EVS:GF#13880](#) - Grid Enablement

The LexBIG API was Grid-enabled as a prototype for the close of the previous Mayo contract. During the 4.2 timeframe Mayo will extend this prototype to provide a more comprehensive of the LexBIG API and deploy to the NCI environment.

[EVS:GF#13881](#) - Expand LexBIG Model

A portion of the LexGRID/LexBIG model was modified and adjusted as necessary to make it compatible with the SDK for the close of the previous Mayo contract. During the 4.2 timeframe Mayo will extend the SDK Compliant Model.

[EVS:GF#13882](#) Silver Level Compatibility

The process of applying for Silver Level Compatibility of the LexBIG API will be initiated. A Silver Level Compatibility Submission Package will be created and submitted for review.

Bugs Fixed Since Last Release

[EVS:GF#8670](#) - MedDRA SMQ attributes not loaded properly

Standardized MedDRA Query (SMQ) attributes were introduced into the 10.0 release of MedDRA. There is currently a problem loading this information to LexBIG as association qualifiers.

[EVS:GF #10439](#)/[EVS:GF #10884](#) MedDRA security needs to be enforced

When searching for concepts against the MedDRA Vocabulary, the Distributed LexBIG API should return null if the user has not input a token using the following provided SecurityToken methods. Instead it is returning results even when no token is sent.

[EVS:GF #14252](#)/[EVS:GF #13808](#) - CL CUI -> CUI Mapping Required

To maintain accessibility by client applications that may store and later attempt to access temporary ids assigned by Metathesaurus, mappings need to be established. A loader needs to be created to load the history information into LexBIG and then the data can be exposed through the LexBIG history service. The EVS API needs to handle the actual interface between the client application and the history service.

[EVS:GF #15437](#) - `getCodingSchemesWithSupportedAssociation` iterations

A `SecurityToken` must be passed in along with the URN to correct problems with this method call.

[EVS:GF #15474](#) - Make EVSAPI compatible to LexBIG 2.3 Release.

The LexEVS API needs to be updated to work with the LexBIG 2.3.0 release.

[EVS:GF #16158](#) - Null point exception: `LexGrid.LexBIG.Impl.function.query.TestAttributePresenceMatch.matchAttribute`.

When running the Junit for EVSAPI we received the following error.

Null point exception- `LexGrid.LexBIG.Impl.function.query.TestAttributePresenceMatch.matchAttribute`.

The test case is wrong in `TestAttributePresenceMatch.matchAttribute` line no-73 needs to be changed.

[EVS:GF #16291](#) - `ResolvedConceptReferencesIterator` returning same concept on calling `next()`

On calling `next()` method of `ResolvedConceptReferencesIterator` when resolving a `CodedNodeSet`, the iterator is returning the same concept again and again. This seems to be happening only with distributed calls. Attaching two programs making distributed and local calls.

[EVS:GF #16405](#) - `MatchRootName` preference not recognized on OWL load

The MGED ontology does not declare 'kinds' and therefore root nodes are not detected the same as historic releases of the NCI Thesaurus. An xml-based preference file was created to detect the single known root node as follows. However, the setting was not recognized.

[EVS:GF #16438](#) - Unable to apply manifest post-load

By design, manifest load disallows change to primary coding scheme identifiers (e.g. coding scheme name and registered name) when applying a manifest to an already loaded scheme. However, the load should not fail if the identifying information matches the already loaded scheme. Currently it is assumed that if the manifest provides the identifiers it is a change and the load fails.

[EVS:GF #16478](#) - Export to LexGrid XML does not recognize menu selection

If using the the command line program and a specific URN and Version are not provided as parameters, the program will prompt for a coding scheme to export. However, the coding scheme choice is not recognized after selection.

[EVS:GF #17063](#) - `EVSQueryDAO` `getHasChildrenbyCode` method fails on NCI Thesaurus root node, Conceptual Entity (C20181).

`EVSQueryDAO` `getHasChildrenbyCode` method fails on NCI Thesaurus root node, Conceptual Entity (C20181). It is because that this concept has two subconcepts both called "Channel" (`getEntityDescription().getContent()`). The sorting algorithm needs to be modified to handle such cases.

[EVS:GF #14485](#) - Semantic Net Loader should use SRSTR for association generator

Currently SRSTRE1 RRF file is being used by the Semantic Net loader to generate associations between the concepts. The associations generated using SRSTRE1 file seems to be very flat and not hierarchical. For example, concept 'Vertebrate' (T010) is associated to 'Organism' (T001), 'Physical Object' (T072), 'Event' (T071) and 'Animal' (T008) as 'isa' association.

RRF file SRSTR seems to be having associations in proper hierarchy. Using this file, concept 'Vertebrate' (T010) is associated only to 'Animal' (T008) as 'isa' association.

So using SRSTR to generate associations between concepts will give a correct hierarchical representation.

[EVS:GF #15015](#) - Problem with matching properties

While testing using the fly anatomy ontology in the GUI, I ran into a problem if I search for the word "lateral" using restrict to matching properties and select all the Match Properties and select all the Property Types. If I unselect `conceptCode` from the list of Match Properties, then things work fine.

[EVS:GF #16155](#) - Xerces `cvc-complex-type.4: Attribute 'isImported' must appear on element 'supportedCodingScheme'`.

While testing 2.3.0.Manifest file for the following

HL7_Manifest.

MGED_MF_1.3.1

NSDFRT_june2006_loadmanifest

Thesaurus-Bycode-07.12e_MF

VA_NDFRT_072606_MF

Zebrafish_MF

we received the following error "Attribute 'isImported' must appear on element 'supportedCodingScheme'.

Known Issues

[EVS:GF #15776](#) - `DLBWrapper.getSupportedCodingSchemes` returns null

The `DLBWrapper.getSupportedCodingSchemes` returns null if any of the codingSchemes in the repository require secure access.

Currently the first vocabulary that it attempts to access is NCI Metathesaurus, which is secured. Remains to be seen if it will return anything if the secured vocab is not first.

[EVS:GF #15817](#) - Inconsistency in the application of codingScheme name in DLBAdapter.

In the `DLBAdapter` you can do a `.setVocabulary()` to set the default codingScheme, version and tag that some queries of the `DLBAdapter` will run against. However, this default codingScheme is not consistently applied.

1. Some methods in DLBAdapter allow you to pass in a codingSchemeName. Of these, some use the passed in codingSchemeName with a null CodingSchemeVersionOrTag. Others use the passed in codingSchemeName plus the version or tag passed in as the default in .setVocabulary.
2. Some methods with no parameters automatically use the defaults. Others don't.

Deferred/Deprecated Items

As the EVS Team looks toward the next major release of LexEVS (v5.0), we have begun to anticipate functionality that will naturally be absorbed by the LexBIG API. Those items have been classified as deprecated in our GForge tracking system. Additionally, those items that have not been evaluated as high priority items, have been deferred to the 5.0 release. These items/trackers can be view [here](#).

CORE Product Dependencies

Refer to the [CORE Product Dependency Matrix](#) for the caCORE SDK and other software technologies on which this release of this product relies.