1 - LexEVS 6.x Installation Overview

Contents of this Page

- Introduction
- Overview of LexEVS Services
- Overview of LexEVS Local Runtime
  - LexEVS Local Runtime Installed components
  - LexEVS Components (Detailed)
    - Service Management
    - LexEVS API
    - Documentation
    - Automated Test Suite
    - Configuration and Resource Files
    - Other Installed Components
  - Deployment Alternatives
- Overview of LexEVS Distributed
- Overview of LexEVS Graph-Resolve REST Service and NodeGraphResolutionExtension
  - NodeGraphResolutionExtension
  - Graph-Resolve Service
- Overview of LexEVS Grid Services (No longer available or Supported – Removed in 6.3)
  - Analytical Grid Service (Deprecated - Removed in 6.3)
  - Data Grid Service (Deprecated - Removed in 6.3)
- LexEVS 6.x CTS2 Specific Installations

LexEVS 6.x Installation Links

- Install Guide Main Page
  - Overview
  - Prerequisites and Platforms
  - Local Runtime
  - Local Runtime Command Line
  - Distributed
  - CTS2 Services
  - URI Resolver Service
- LexEVS 6.0 Main Page
- LexEVS Current Release

Introduction

This document is a section of the Installation Guide. It provides the user with an overview of services provided by the installed LexEVS runtime.

Overview of LexEVS Services

The LexEVS package represents a comprehensive set of software and services to author, publish, and access vocabulary (and to some extent, do so in a variety of web-enabled and grid environments.) Cancer Centers can use the LexEVS package set to install the NCI Thesaurus, NCI Metathesaurus content as well as other terminologies, and provide query access via a rich application programming interface (API). LexEVS services can be used in numerous applications wherever vocabulary content is needed. Content can be loaded from a number of formats including but not limited to OWL, OBO, and RRF.

LexEVS is intended to address the needs of the following groups:

- **Vocabulary service providers:** Describes organizations currently supporting externalized API-level interfaces to vocabulary content.
- **Vocabulary integrators:** Describes organizations that desire to integrate new vocabulary content.
- **Vocabulary users:** Describes those interested in utilizing vocabulary services.
- **Vocabulary authors:** Describes those interested in authoring vocabularies.

Overview of LexEVS Local Runtime

LexEVS Local Runtime Installed components

1. Service Management consists of an API and example programs to load, index, publish, and manage both vocabulary content and service metadata for the vocabulary server. Examples are found in the `admin` directory of the LexEVS root installation directory.
2. LexEVS Application Programming Interface (API) is comprised of methods to support Lexical Operations, Graph Operations, Value Set Operations, Authoring Operations, and History Operations. Examples are found in the `examples` directory of the LexEVS root installation directory.
3. Documentation consists of detailed JavaDocs and the LexEVS 6.0 Programmer's Guide and LexEVS 6.x Administration Guide.

4. Automated Test Suite is provided to validate the LexEVS installation. It includes relatively small sample vocabularies that can be loaded by the user. These can be found in the `test` directory and its sub-directory, `testData`.
5. Configuration and Resource Files provide custom configuration of the LexEVS runtime to users.

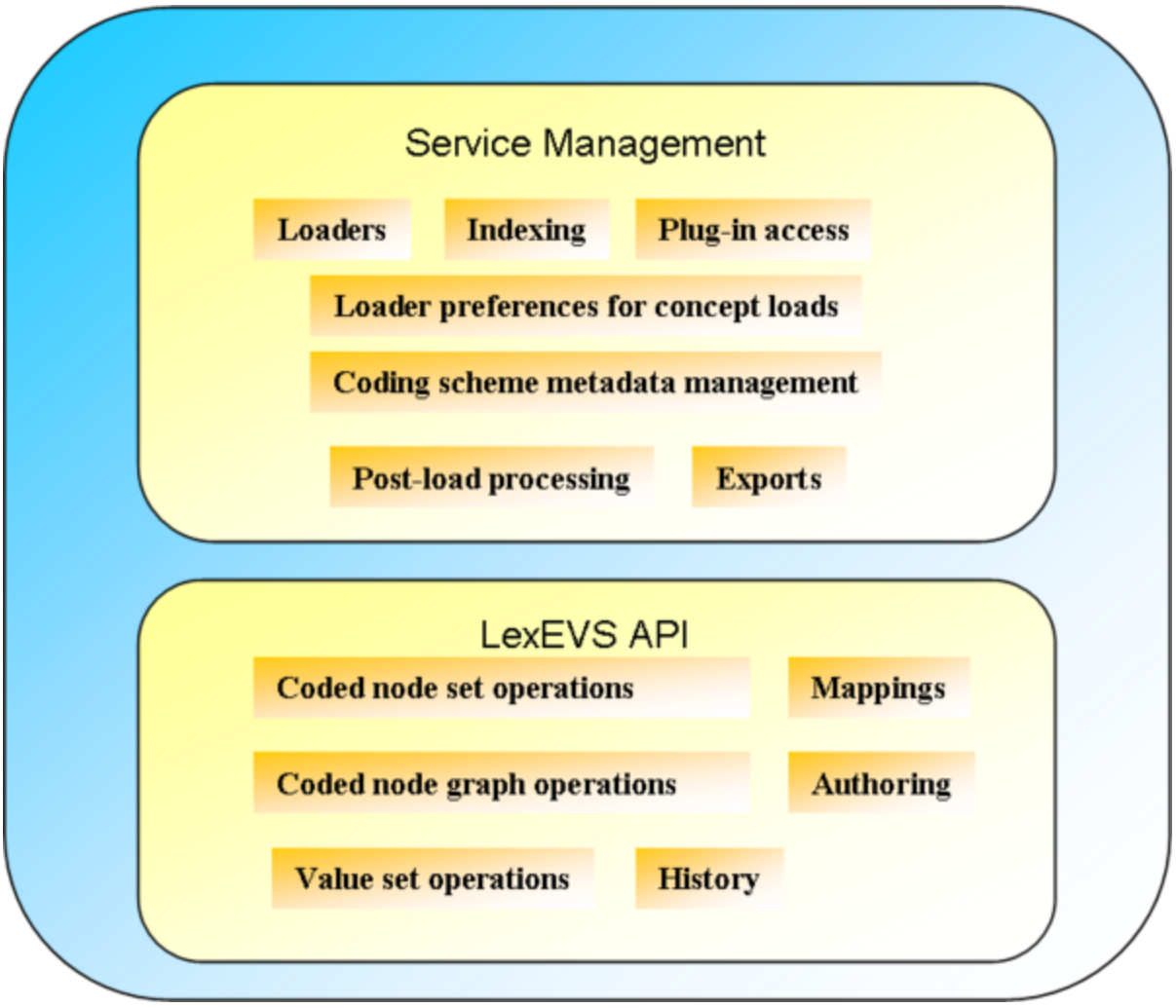
Service Management comprises the following components:

- Loaders
- Indexing
- Plug-in access
- Loader preferences for concept loads
- Coding Scheme Metadata Management
- Post-load processing
- Exports

LexEVS API comprises the following components:

- Coded net set operations
- Mappings
- Coded net graph operations
- Authoring
- Value set operations
- History

The following figure shows the installed services and API.



Note

Additional information about the LexEVS API is provided in the LexEVS 6.0 Programmer's Guide. Information related to Service Management of a LexEVS Terminology Service can be found in the LexEVS 6.x Administration Guide.

LexEVS Components (Detailed)

The LexEVS installation includes the following components:

Service Management

Administrative Programs for managing LexEVS service and available in the `admin` directory:

- **Loading**– Loading extensions to LexEVS accessed through a Service Management API that supply the following:
  - Terminology Loads from various formats:
  - Loading OWL, OBO, RRF and native LexGrid XML as well as less universal formats
    - Examples Available in the `admin` directory: LoadOWL, LoadOBO, LoadNCIThesOWL
  - Manifest Loads that load Coding Scheme metadata changes to an terminology in the service
    - Example: LoadManifest or as -mf option to most coding scheme loads.
  - Supplemental Coding Scheme Loads that load supplements to existing coding schemes.
    - Example: Load option for a coding scheme. Currently only available as an option in the lbGUI application.
  - Revision Loads that load changes to any given coding scheme element using a LexEVS XML schema compliant document, pushing these changes through a revision engine.
    - Example: LoadLgXML using the revision XML entry point to LexGrid XML.
  - History Loads that load histories of coding scheme changes, currently specific to NCI Thesaurus and Metathesaurus.
    *(Mappings between coding schemes loaded from MRMAP and MRSAT RRF files, or from LexGrid XML.
  - Value Set and Pick List Loads:
    - Examples: LoadLgXML using ValueSetDefinition or PickListDefinition as entry points in a LexGrid XML document.
    - Examples: SourceAssertedValueSetDefinitionLoad
  - Graphing Database loads for high speed resolution of terminology hierarchies and relationships
    - LoadGraphDataBase
  - Map Loads for mapping between terminologies.
    - Example: LoadMrMap or LoadLgXML
- **Exporting**– Exporting any terminology in a LexEVS terminology service to one of three formats:
  - Export to OWL
  - Example: ExportOWL
  - Export to OBO
  - Example: ExportOBO
  - Export to LexGrid XML
  - Example: ExportLgXML
- **Indexing**– normally takes place during a terminology load, but re-indexing using the Service Management API is possible:
  - Examples: RemoveIndex, RebuildIndex, BuildAssertedValueSetIndex
- **Coding Scheme Metadata Management:**Loading Manifests to adjust most elements of coding scheme metadata.
  - Examples: See loading manifest for metadata changes above or examples ActivateScheme, TagScheme in the admin directory.
- **Plug-in access:**All plug-ins developed for LexEVS are accessed through Service Management.
  - Examples: LexEVSService.getExtension(<plug-in class name>);
- **Loader Preferences for entity(concept) loads**– Customization of how entities are loaded allows users to make choices as to how a terminology appears in LexEVS.
  - Available during load time only and as option -lp to most loaders
- **Post Load Processing**– Specialty processing options for post load, such as providing the approximate number of concepts examples in the source for the lbGUI application.
  - Example for post processor implementation is in org.LexGrid.LexBIG.gui.load.PostProcessorLauncher using the LoaderPostProcessor class.

LexEVS API

Including Program Examples for common vocabulary functions using sample vocabulary and Coded Node functions found in the `examples` directory

- **Coded Node or Lexical Set Operations:**Set operations for coded entities which group, query and apply set functions to sets of coded entries.
  - Query Functions: Based on restrictions to a set of coded nodes retrieved from a given coding scheme a list of nodes is returned:
  - Examples:
    - FindCodesForDescription
    - SoundsLike
  - Set Operations: sets of coded nodes have set operations performed upon them.
  - Union
  - Intersection
  - Difference
- **Coded Node Graph Operations:**Based on a set of restrictions, including types of relationships, a coded node graph is returned.
  - Examples:
    - FindPropsAndAssocForCode
    - FindRelatedCodes
    - FindTreeForCodeAndAssoc
- **Value Set Operations:**Operations on Value Sets and Pick Lists.
  - Examples:
    - CheckForConceptInValueSet
    - CheckForValueSetSubSet
- **Source Asserted Value Set Operations**: Operations on Value Sets Asserted in the Source Terminology
  - Examples:
    - listAllSourceAssertedValueSets
    - getSourceAssertedValueSetForValueSetURI
- **Authoring Operations:** Creating Coding Schemes and all subelments
- **Mapping Operations:** Creating mappings (a part of Coding Scheme creation)

Documentation

- **JavaDocs**
- **Links to:**
  - LexEVS 6.x Programmer's Guide
  - LexEVS 6.x Installation Guide
  - LexEVS 6.x Administration Guide

Automated Test Suite

A shell script providing verification of a correct installation of LexEVS.

Configuration and Resource Files

Contain configuration files and post terminology load indexes servicing LexEVS terminologies – located in the `resource_ directory`.

- **Configuration files:** enable you to customize your installation to meet your specific database, server, and other network needs and are located in a resource subfolder `config`;
  - lbconfig.props: This file *must be* edited to at least provide a database connection for LexEVS.
- **Post Load Indexes:** Lucene indexes in the resource subfolder `lbIndex`.
  - These indexes are readable by the LexEVS api or a Lucene index browser like Luke.

Other Installed Components

- **Runtime:** contains the combined LexEVS classes and third party dependencies required for LexEVS to function in a single archive jar file.
- **Runtime components:** a combined archive of LexEVS classes but with separate 3rd party jars outside of archive.
- **Unistaller:** a packaged uninstaller, that just deletes the root directory. Database cleanup is left up to the user.

Deployment Alternatives

Earlier versions of the LexEVS local Runtime package offered flexible database deployment alternatives depending on the underlying data base management systems(DBMS's). Single database configurations allow the user to manage databases more effectively so LexEVS no longer offers the option of loading each terminology as a separate database.

Deploying using single table or multiple table mode are the current options. Data is either collated to a single schema table or broken up by terminologies for a table of each kind for each terminology. Single table modes can be very resource intensive when deleting a large terminology such as the NCI Metathesaurus.

Overview of LexEVS Distributed

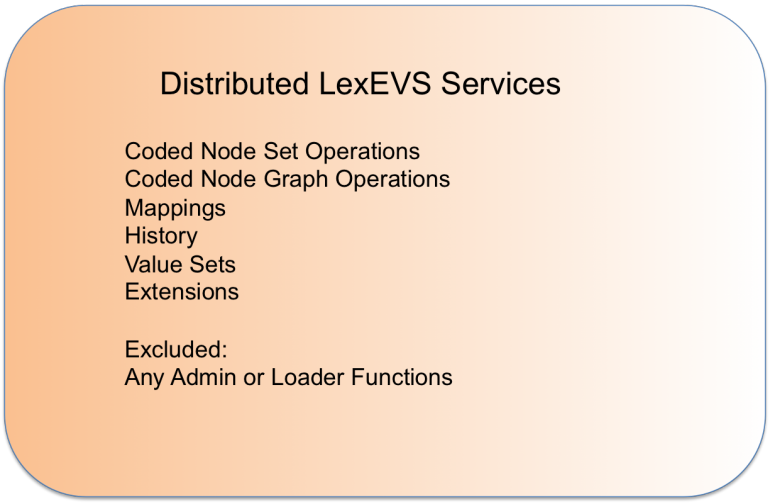
(No longer available to the public as a service from NIH/NCI) It is still possible to run and install this service yourself, but it is being sunsetted as an application and will eventually be deprecated. See the CTS2 service below.

The Distributed LexEVS environment is distributed as a single WAR archive designed to be installed in a web application container.

Distributed Installed Components

- **A Subset of the LexEVS API:** Management, administration and authoring tools are not included for Java remote method invocation (RMI). Central querying functionality is the emphasis.
- **Query By Example Querying Services (Deprecated - Removed in 6.5):** A Java based version of the database querying interface that makes queries based on calls defined by LexGrid model elements.
- **Web Services for SOAP messaging (Deprecated - Removed in 6.5, Replaced by the CTS2 Service):** Providing a platform independent service to the database for LexGrid model elements to users of Perl, .net, C/C++, C#, Python and so on.
- **XML-HTTP or REST Services(Deprecated - Removed in 6.5, Replaced by the CTS2 Service):** Providing LexGrid Model query access to any user with a browser or browser like application.

The following figure shows Distributed Remote LexEVS JAVA API with Coded Node Set Operations, Coded Node Graph Operations, Mappings, and History.



Overview of LexEVS Graph-Resolve REST Service and NodeGraphResolutionExtension

Provides a graphing database REST Service wrapped by a LexEVS extension compatible API in java.

A SpringBoot enabled REST service queries an Arangodb instance for graph resolutions originally loaded from the LexEVS associations tables.

NodeGraphResolutionExtension

A set of methods designed to allow high speed resolution of larger graphs insuring a knowledge of graph resolution result sizes in a timely manner.

Graph-Resolve Service

A REST service for Non-LexEVS users that provides a platform independent format for resolved graph elements.

Overview of LexEVS Grid Services (No longer available or Supported – Removed in 6.3)

Separate Analytical and Data Services are distributed in war archives for installation in web application containers.

Analytical Grid Service (Deprecated - Removed in 6.3)

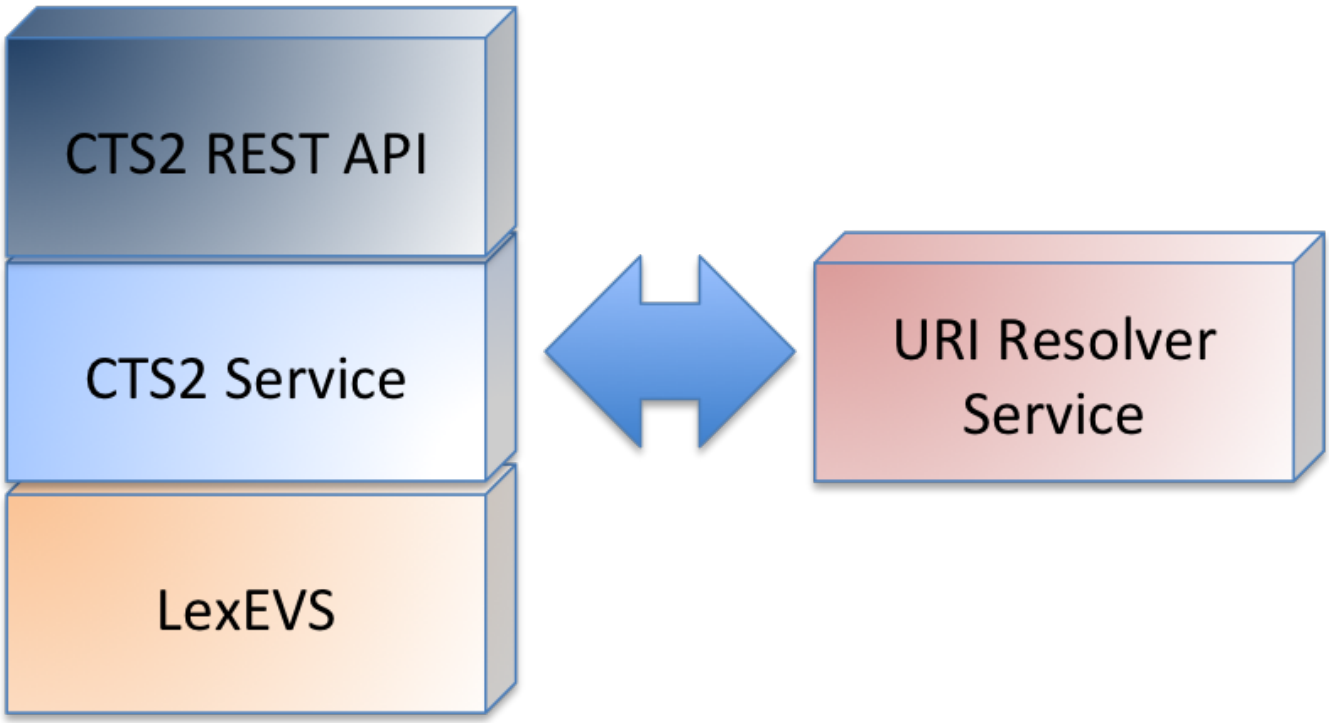
Provides the same subset of the LexEVS APIs as the Distributed environment but is ISO 21090 compliant. The following figure shows the LexEVS Grid Service Java API with Coded Node Set Operations, Coded Node Graph Operations, Mappings, and History.

Data Grid Service (Deprecated - Removed in 6.3)

Allows users to make federated queries, using the CQL XML based query language, of LexGrid model elements hosted at LexEVS grid service nodes. The following figure shows a Federated CQL Queries element.

LexEVS 6.x CTS2 Specific Installations

LexEVS 6.1 features a CTS2 Compatible REST API. The REST server and an accompanying URI resolution service are installed to support this service.


CTS2 REST API
CTS2 Service
LexEVS
URI Resolver Service