

# 1 - LexEVS 5.x Installation Overview

## Contents of this Page

- [Introduction](#)
- [Overview of LexEVS Services](#)
- [LexEVS Components \(Overview\)](#)
- [LexEVS components \(Detailed\)](#)
- [Deployment Alternatives](#)

## Introduction

This document is a section of the [LexEVS 5.x Installation Guide](#).

## Overview of LexEVS Services

The LexEVS package set represents a compressive set of software and services to load, publish, and access vocabulary (and do so in a variety of web-enabled and grid environments.) Cancer Centers can use the LexEVS package set to install NCI Thesaurus and NCI Metathesaurus content queryable via a rich application programming interface (API). LexEVS services can be used in numerous applications wherever vocabulary content is needed.

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

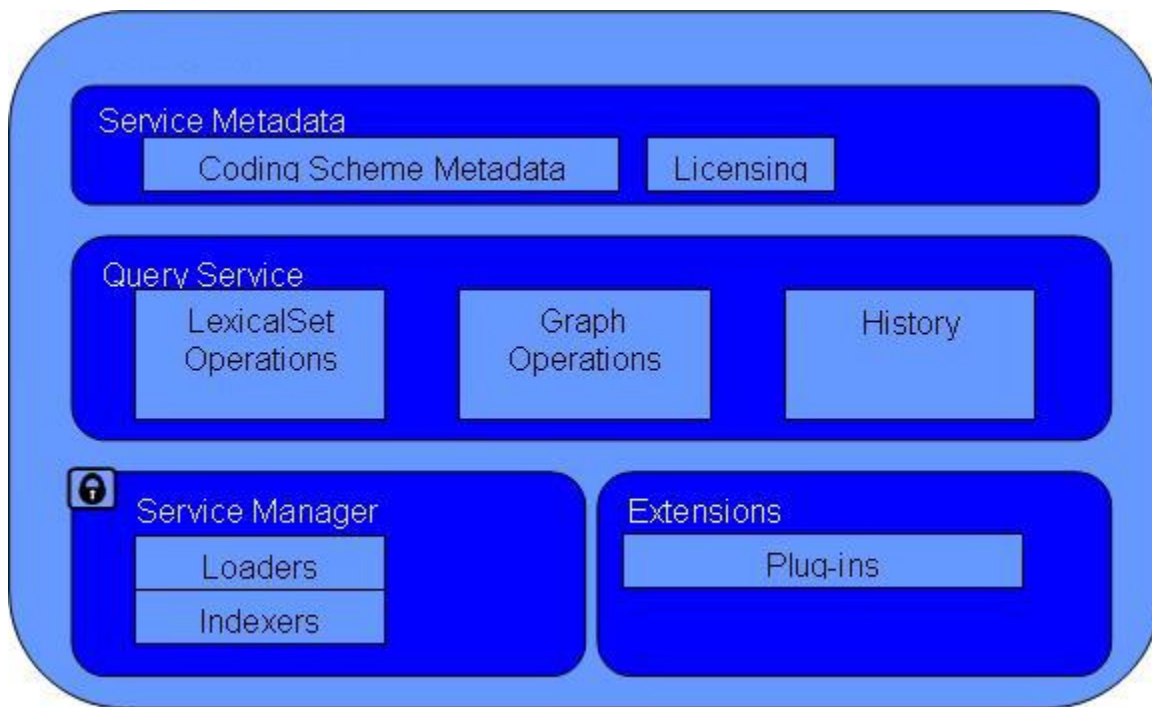
- *Vocabulary service providers.* Describes organizations currently supporting externalized API-level interfaces to vocabulary content for the caBIG® community.
- *Vocabulary integrators.* Describes organizations within the caBIG® community that desire to integrate new vocabulary content to be served to the caBIG® community.
- *Vocabulary users.* Describes the caBIG® community interested in utilizing vocabulary services within the context of other caBIG® projects.

## LexEVS Components (Overview)

1. **Service Management** consists of programs to load, index, publish, and manage vocabulary content for the vocabulary server.
2. **Application Programming Interface (API)** is comprised of methods to support Lexical Operations, Graph Operations, and History Operations.
3. **Documentation** consists of detailed JavaDocs and Programmers Guide.
4. **Examples** are provided as sample source code for common vocabulary queries.
5. **Test Suite** is provided to validate the LexEVS installation.

The following graphic shows LexEVS service components:

- Service Metadata which comprises the Metadata coding scheme and Licensing.
- Query Service which comprises Lexical Set Operations, Graph Operations, and History.
- Service Manager which comprises Loaders and Indexers.
- Extensions which comprises Plug-ins.



#### Note

Additional information about the LexEVS API is provided in the [WIP Format LexEVS 5.x Programmer's Guide](#).

## LexEVS components (Detailed)

The LexEVS installation includes the following components:

- **Service Manager** or Administrative Programs for managing LexEVS server including Loaders, Indexers and Service Metadata Queries.
  - ActivateScheme
  - ClearOrphanedResources
  - CodingSchemeSelectionMenu
  - DeactivateScheme
  - ExportLgXML
  - ExportOBO
  - ExportOWL
  - ListExtensions
  - ListSchemes
  - LoadFMA
  - LoadHL7RIM
  - LoadLgXML
  - LoadMetadata
  - LoadNCIHistory
  - LoadNCIMeta
  - LoadNCIThesOWL
  - LoadOBO
  - LoadOWL
  - LoadRadLex
  - LoadUMLSDatabase
  - LoadUMLSFiles
  - LoadUMLSHistory
  - LoadUMLSSemnet
  - RebuildIndex
  - RemoveIndex
  - RemoveMetadata
  - RemoveScheme
  - TagScheme
  - TransferScheme
- **Query Service** including Program Examples for common vocabulary functions using sample vocabulary and CodedNodeSet functions
  - **Lexical Set Operations**
    - FindCodesForDescription
    - SoundsLike
    - Union
    - Intersection

- Difference
- **Graph Operations**
  - FindPropsAndAssocForCode
  - FindRelatedCodes
  - FindTreeForCodeAndAssoc
- LexEVS Automated Verification Test Suite
- LexEVS Runtime jar (combined archive)
- LexEVS Runtime components (combined archive with 3rd party jars outside of archive)
- LexEVS Uninstaller
- LexEVS License Terms and Conditions
- Configuration files to enable you to customize your installation to meet your specific database, server, and other network needs
  - llbconfig.props
- Documentation
  - JavaDocs
  - Links to:
    - LexEVS Programmer's Guide
    - LexEVS Installation Guide

## Deployment Alternatives

The LexEVS local Runtime package has flexible database deployment alternatives depending on the underlying dbms. For dbms's like MySQL and PostGres the user can deploy terminologies contained in a single database or in multiple databases. Single database configurations allow the user to manage databases more effectively. Multiple data base configurations provide a little more transparency to the underlying terminology load. Some dbms's, like Oracle, require the single data base configuration.