

LexEVS 5.x Vocabulary Administration

Contents of this Page

- [Introduction](#)
- [Vocabulary Administration Overview](#)
- [Installing Sample Vocabularies](#)
- [Running the Sample Query Programs](#)
- [Installing NCI Vocabularies](#)
 - [NCI Thesaurus Vocabulary](#)
 - [NCI Metathesaurus Vocabulary](#)
 - [NCI History](#)
- [Deactivating and Removing a Vocabulary](#)
- [Tagging a Vocabulary](#)

Introduction

This document is a section of the [Administration Guide](#).

Vocabulary Administration Overview

A set of administrative utilities are provided to manage the LexEVS Service. These utilities are provided for Windows (.bat) and Linux (.sh) operating systems. Each of the commands is located in the {LEXBIG_DIRECTORY}/admin and {LEXBIG_DIRECTORY}/test directory. A full description of the options with example is provided for each of the administration utilities.

Administrative Program	Description
ActivateScheme	Activates a coding scheme based on unique URN and version. Options: • -u,-urn <urn> URN uniquely identifying the code system. • -v,-version <versionId> Version identifier. • -f,-force Force activation (no confirmation). Example: <pre>ActivateScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.09e"</pre>
ClearOrphanedResources	Clean up orphaned resources - databases and indexes. Options: • -li,-listIndexes List all unused indexes. • -ldb,-listDatabases List all unused databases (with matching prefix). • -ri,-removeIndex <name> Remove the (unused) index with the given name. • -rdb,-removeDatabase <name> Remove the (unused) database with the given name. • -a,-all Remove all unreferenced indexes and databases (with matching prefix). Example: <pre>ClearOrphanedResources -li</pre>
DeactivateScheme	Deactivates a coding scheme based on unique URN and version. Options: • -u,-urn <urn> URN uniquely identifying the code system. • -v,-version <versionId> Version identifier. • -d,-date <yyyy-MM-dd.HH:mm:ss> Date and time for deactivation to take effect; immediate if not specified. • -f,-force Force deactivation (no confirmation). Example: <pre>DeactivateScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.09e" -d "01/31/2099,12:00:00"</pre>

ExportLgXML	Exports content from the repository to a file in the LexGrid canonical XML format. Options: • -out,-output <uri> URI or path of the directory to contain the resulting XML file. The file name will be automatically derived from the coding scheme name. • -u,-urn <name> URN or local name of the coding scheme to export. • -v,-version <id> The assigned tag/label or absolute version identifier of the coding scheme. • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors • -f,-force If specified, allows the destination file to be overwritten if present.  Note If the coding scheme and version values are unspecified, a list of available coding schemes will be presented for user selection. Examples: <pre>ExportLgXML -out "file:///path/to/dir" -nf -f</pre> <pre>ExportLgXML -out "file:///path/to/dir" u "NCI_Thesaurus" -v "PRODUCTION" -nf -f</pre>
ExportOBO	Exports content from the repository to a file in the Open Biomedical Ontologies (OBO) format. Options: • -out,-output <uri> URI or path of the directory to contain the resulting OBO file. The file name will be automatically derived from the coding scheme name. • -u,-urn <name> URN or local name of the coding scheme to export. • -v,-version <id> The assigned tag/label or absolute version identifier of the coding scheme. • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors • -f,-force If specified, allows the destination file to be overwritten if present.  Note If the coding scheme and version values are unspecified, a list of available coding schemes will be presented for user selection. Examples: <pre>ExportOBO -out "file:///path/to/dir" -nf -f</pre> <pre>ExportOBO -out "file:///path/to/dir" -u "FBbt" -v "PRODUCTION" -nf -f</pre>
ExportOWL	Exports content from the repository to a file in OWL format. Options: • -out,-output <uri> URI or path of the directory to contain the resulting OWL file. The file name will be automatically derived from the coding scheme name. • -u,-urn <name> URN or local name of the coding scheme to export. • -v,-version <id> The assigned tag/label or absolute version identifier of the coding scheme. • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors • -f,-force If specified, allows the destination file to be overwritten if present.  Note If the URN and version values are unspecified, a list of available coding schemes will be presented for user selection. Example: <pre>ExportOWL -out "file:///path/to/dir" -nf -f</pre> <pre>ExportOWL -out "file:///path/to/dir" -u "sample" -v "1.0" -nf -f</pre>

ListExtensions	List registered extensions to the LexEVS runtime environment. Options: • -a, -all List all extensions (default, override by specifying other options). • -i, -index List index extensions. • -m, -match List match algorithm extensions. • -s, -sort List sort algorithm extensions. • -g, -generic List generic extensions. Example: <pre>ListExtensions -a</pre>
ListSchemes	List all currently registered vocabularies. Options: • -b, -brief List only coding scheme name, version, urn, and tags (default). • -f, -full List full detail for each scheme. Example: <pre>ListSchemes</pre>
LoadLgXML	Loads a vocabulary file, provided in LexGrid canonical xml format. Options: • -in, -input <uri> URI specifying location of the source file. • -v, -validate <level> Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: ◦ 0 = Verify document is well-formed ◦ 1 = Verify document is valid • -nf, -noFail If specified, indicates that processing should not stop for recoverable errors. • -a, -activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated. • -t, -tag <tagID> An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. Load Example: <pre>LoadLgXML -in "file:///path/to/file.xml" -nf -a</pre> Validation Example: <pre>LoadLgXML -in "file:///path/to/file.xml" -v 0</pre>

LoadNCIHistory	Imports NCI History data to the LexEVS repository. Options: • -in, -input <uri> URI specifying location of the history file • -vf, -versionFile <uri> URI specifying location of the file containing version identifiers for the history to be loaded. • -v, -validate <level> Perform validation of the candidate resource without loading data. If specified, the '-nf' and '-r' options are ignored. Supported levels of validation include: ◦ 0 = Verify top 10 lines are correct format ◦ 1 = Verify correct format for the entire file • -nf, -noFail If specified, indicates that processing should not stop for recoverable errors • -r, -replace If not specified, the provided history file will be added into the current history database; otherwise the current database will be replaced by the new content. Load Example: <pre>LoadNCIHistory -nf -in "file:///path/to/history.file" -vf "file:///path/to/version.file"</pre> Validation Example: <pre>LoadNCIHistory -in "file:///path/to/history.file" -v 0</pre> Versions File Format Information: <pre>releaseDate isLatest releaseAgency releaseId releaseOrder entityDescription</pre> Sample record: <pre>28-NOV-05 false http://nci.nih.gov/ 05.10e 26 Editing of NCI Thesaurus 05.10e was completed on October 31, 2005. Version 05.10e was October's fifth build in our development cycle.</pre>
LoadNCIMeta	Loads the NCI MetaThesaurus, provided as a collection of RRF files. Options: • -in, -input <uri> The directory containing the RRF files; in URI format. • -v, -validate <level> Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: ◦ 0 = Verify first 1000 lines per required file • -nf, -noFail If specified, indicates that processing should not stop for recoverable errors • -a, -activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated • -t, -tag <tagID> An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. Load Example: <pre>LoadNCIMeta -in "file:///path/to/directory" -nf -a</pre> Validation Example: <pre>LoadNCIMeta -in "file:///path/to/directory" -v 0</pre>

LoadNCIThesOWL	Loads an OWL file containing a version of the NCI Thesaurus ... Options: • -in,-input <uri> URI specifying location of the source file • -v,-validate <level> Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: ◦ 0 = Verify document is well-formed ◦ 1 = Verify document is valid • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors. • -a,-activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated. • -t,-tag <tagID> An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. Load Example: <pre>LoadNCIThesOWL -in "file:///path/to/thesaurus.owl" -nf -a</pre> Validation Example: <pre>LoadNCIThesOWL -in "file:///path/to/thesaurus.owl" -v 0</pre>
LoadOBO	Loads a file specified in the Open Biomedical Ontologies (OBO) format. Options: • -in,-input <uri> URI or path specifying location of the source file • -v,-validate <int> Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: ◦ 0 = Verify document is valid • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors • -a,-activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated • -t,-tag <id> An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. Example: <pre>LoadOBO -in "file:///path/to/file.obo" -nf -a</pre> <pre>LoadOBO -in "file:///path/to/file.obo" -v 0</pre>
LoadOWL	Loads an OWL file.  Note Load of the NCI Thesaurus should be performed via the LoadNCIThesOWL counterpart, since it will allow more precise handling of NCI semantics. Options: • -in,-input <uri> URI or path specifying location of the source file • -v,-validate <int> Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: ◦ 0 = Verify document is well-formed ◦ 1 = Verify document is valid • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors • -a,-activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated • -t,-tag <id> An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. Examples: <pre>LoadOWL -in "file:///path/to/somefile.owl" -nf -a</pre> <pre>LoadOWL -in "file:///path/to/somefile.owl" -v 0</pre>

LoadUMLSDatabase	Loads UMLS content, provided as a collection of RRF files in a single directory. Files may comprise the entire UMLS distribution or pruned via the MetamorphoSys tool. A complete list of source vocabularies is available online at http://www.nlm.nih.gov/research/umls/metaa1.html . Options: • -in,-input <uri> Location of the source database. Typically this is specified in the form of a URL that indicates the database server, port, name, and optional properties. • -u,-uid User ID for authenticated access, if required and not specified as part of the input URL. • -p,-pwd Password for authenticated access, if required and not specified as part of the input URL. • -d,-driver Name of the JDBC driver to use when accessing the database. • -s,-sources Comma-delimited list of source vocabularies to load. If absent, all available vocabularies are loaded. • -v,-validate <int> Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: ◦ 0 = Verify the existence of each required file • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors • -a,-activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated. • -t,-tag <id> An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. Examples: <pre>LoadUMLSDatabase -in "jdbc:postgresql://localhost:5432/lexgrid" -d "org.postgresql.Driver" -u "myDatabaseUser" -p "myPassword" -s "ICD9CM_2005,ICD9CM_2006" -nf -a</pre> <pre>LoadUMLSDatabase -in "jdbc:postgresql://localhost:5432/lexgrid" -d "org.postgresql.Driver" -u "myDatabaseUser" -p "myPassword" -v 0</pre>
LoadUMLSFiles	Loads UMLS content, provided as a collection of RRF files in a single directory. Files may comprise the entire UMLS distribution or pruned via the MetamorphoSys tool. A complete list of source vocabularies is available online at http://www.nlm.nih.gov/research/umls/metaa1.html . Options: • -in,-input <uri> URI or path of the directory containing the NLM files • -s,-sources Comma-delimited list of source vocabularies to load. If absent, all available vocabularies are loaded. • -v,-validate <int> Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: ◦ 0 = Verify the existence of each required file • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors • -a,-activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated. • -t,-tag <id> An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. Examples: <pre>LoadUMLSFiles -in "file:///path/to/directory/" -s "ICD9CM_2005,ICD9CM_2006" -nf -a</pre> <pre>LoadUMLSFiles -in "file:///path/to/directory/" -v 0</pre>  Note UMLS Metathesaurus RRF files are a very large filesset. Many users prefer to subset these files using the Metamorphosys tool included with the UMLS Metathesaurus in order to move a single terminology from a central location of these files. When generating source RRF files from the Metathesaurus, the Metamorphosys tool should be set to output versionless source abbreviations rather than versioned source abbreviations. Failing to do so before loading RRF files to LexEVS will cause an incomplete database to be created leaving the association and concept tables empty

LoadUMLSSemnet	Loads the UMLS Semantic Network, provided as a collection of files in a single directory. The following files are expected to be provided from the National Library of Medicine (NLM) distribution: • LICENSE.txt (text from distribution terms and conditions) • SRFIL.txt (File Description) • SRFIL.txt (Field Description) • SRDEF.txt (Basic information about the Semantic Types and Relations) • SRSTR.txt (Structure of the Network) • SRSTRE1.txt (Fully inherited set of Relations (Uls)) • SRSTRE2.txt (Fully inherited set of Relations (Names)) • SU.txt (Unit Record) These files can be downloaded from the NLM web site at http://semanticnetwork.nlm.nih.gov/Download/index.html . Options: • -in,-input <uri> URI or path of the directory containing the NLM files • -v,-validate <int> Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: ◦ 0 = Verify the existence of each required file • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors • -a,-activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated. • -t,-tag <id> An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. • -il -InheritanceLevel <int> If specified, indicates the extent of inherited relationships to import. 0 = none; 1 = all; 2 = all except is_a (default). All direct relationships are imported, regardless of option. Examples: <pre>LoadUMLSSemnet -in "file:///path/to/directory/" -nf -a -il 1</pre> <pre>LoadUMLSSemnet -in "file:///path/to/directory/" -v 0</pre>
LoadFMA	Imports from an FMA database to a LexEVS repository. Requires that the pprj file be configured with a database URN, username, password for an FMA MySQL based database. The FMA.pprj file and MySQL dump file are available at http://sig.biostr.washington.edu/projects/fm/ upon registration. Options: • -in,-input <uri> URI or path specifying location of the source file • -v,-validate <int> Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: ◦ 0 = Verify document is well-formed ◦ 1 = Verify document is valid • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors • -a,-activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated • -t,-tag <id> An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. Example: <pre>LoadFMA -in "file:///path/to/FMA.pprj" -nf -a</pre> <pre>LoadFMA -in "file:///path/to/FMA.pprj" -v 0</pre>
LoadHL7RIM	Converts an HL7 RIM MS Access database to a LexGrid database • -in,-input <uri> URI or path specifying location of the source file • -mf,-manifest <uri> URI or path specifying location of the manifest file • -lp,-load preferences <uri> URI or path specifying location of the load preferences file • -v,-validate <int> Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: ◦ 0 = Verify document is valid • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors • -a,-activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated • -t,-tag <id> An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. Examples: <pre>LoadHL7RIM -in "file:///path/to/file.mdb" -nf -a</pre> <pre>LoadHL7RIM -in "file:///path/to/file.mdb" -v 0</pre>

LoadMetadata	Loads optional XML-based metadata to be associated with an existing coding scheme. Options: • -u,-urn <name> URN uniquely identifying the code system. • -v,-version <id> Version identifier. • -in,-input <uri> URI or path specifying location of the XML file. • -v,-validate <int> Perform validation of the input file without loading data. If specified, the '-nf', '-f', and '-o' options are ignored. Supported levels of validation include: ◦ 0 = Verify document is valid • -o,-overwrite If specified, existing metadata for the code system will be erased. Otherwise, new metadata will be appended to existing metadata (if present). • -f,-force Force overwrite (no confirmation). • -nf,-noFail If specified, indicates that processing should not stop for recoverable errors  Note If the URN and version values are unspecified, a list of available coding schemes will be presented for user selection. Examples: LoadMetadata -in "file:///path/to/file.xml" -nf -o LoadMetadata -in "file:///path/to/file.xml"
RebuildIndex	Rebuilds indexes associated with the specified coding scheme. Options: • -u,-urn <urn> URN uniquely identifying the code system. • -v,-version <versionId> Version identifier. • -i,-index <name> Name of the index extension to rebuild (if absent, rebuilds all built-in indices and named extensions). • -f,-force Force clear (no confirmation). Example: RebuildIndex -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.09e" -i "myindex"
RemoveIndex	 Note Built-in indices required by the LexEVS runtime cannot be removed. Options: • -u,-urn <urn> URN uniquely identifying the code system. • -v,-version <versionId> Version identifier. • -i,-index <name> Name of the index extension to clear. • -f,-force Force clear (no confirmation). Example: RemoveIndex -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.09e" -i "myindex"
RemoveScheme	Removes a coding scheme based on unique URN and version. Options: • -u,-urn <urn> URN uniquely identifying the code system. • -v,-version <versionId> Version identifier. • -f,-force Force deactivation and removal without confirmation. Example: RemoveScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.09e"

TagScheme	Associates a tag ID (e.g. 'PRODUCTION' or 'TEST') with a coding scheme URN and version. Options: • -u,-urn <urn> URN uniquely identifying the code system. • -v,-version <versionId> Version identifier. • -t,-tag The tag ID (e.g. 'PRODUCTION' or 'TEST') to assign. Example: <pre>TagScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v 05.09e" -t "TEST"</pre>
TestRunner	Located in {LEXBIG_DIRECTORY}/test. Executes a suite of tests for the LexEVS installation.  Note The LexEVS runtime and database environments must still be configured prior to invoking the test suite. Options: • -b,-brief Runs the LexEVS test suite and produce a text report with overall statistics and details for failed tests only. • -f,-full Runs the LexEVS test suite and produce an itemized list of all tests with indication of success/failure. • -h,-html Runs the LexEVS test suite and produce a report suitable for view in a standard web browser. • -x,-xml Runs the LexEVS test suite and produce a report with extensive information for each test case in xml format. Example: <pre>TestRunner -f -h</pre>
TransferScheme	Tool to help gather information necessary to transfer data from one SQL server to another. Options: • -u,-urn The Coding Scheme URN or local name to transfer. • -v,-version The version of the coding scheme to transfer. Example: <pre>TransferScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v 05.09e"</pre>

Installing Sample Vocabularies

This LexEVS installation provides the UMLS Semantic Net and a sampling of the NCI Thesaurus content (sample.owl) that can be loaded into the database.

1. In a **Command Prompt** window, enter the following to go to the example programs:

```
cd {LEXBIG_DIRECTORY}/examples
```

2. To load the example vocabularies, run the appropriate LoadSampleData script (LoadSampleData.bat for Windows; LoadSampleData.sh for Linux).



Note

Vocabularies should not be loaded until configuration of the LexEVS runtime and database server are complete.

The following figure displays the successful load of the sample vocabulary file.

```
C:\WINDOWS\system32\cmd.exe
ing on InputStreamReader or FileReader does not match that of XML document. Use
FileInputStream. [windows-1252 != UTF]
[ProtegeOWLParser] Triple 10000
[ProtegeOWLParser] Triple 20000
[ProtegeOWLParser] Completed triple loading after 9013 ms
[ProtegeOWLParser] Checking untyped resources took 20 ms
[TripleChangePostProcessor] Completed lists after 0 ms
[TripleChangePostProcessor] Completed anonymous classes after 20 ms
[TripleChangePostProcessor] Completed deprecated classes after 10 ms
[TripleChangePostProcessor] Completed properties after 60 ms
[TripleChangePostProcessor] Completed named classes after 80 ms
... Loading completed after 9404 ms
[LexBIG] Total Classes= 2023
[LexBIG] Processing TOP Node... Anatomy_Kind
[LexBIG] Clearing target of NCI_Thesaurus...
[LexBIG] Writing NCI_Thesaurus to target...
[LexBIG] Finished loading DB - loading transitive expansion table
[LexBIG] ComputeTransitive - Processing Anatomic_Structure_is_Physical_Part_of
[LexBIG] ComputeTransitive - Processing hasSubtype
[LexBIG] Finished building transitive expansion - building index
[LexBIG] Indexed 0 concepts.
[LexBIG] Indexed 1000 concepts.
[LexBIG] Indexed 2000 concepts.
[LexBIG] Closing Indexes Sun, 26 Feb 2006 20:54:12
C:\lexbig\admin>
```

Running the Sample Query Programs

A set of sample programs are provided in the {LEXBIG_DIRECTORY}/examples directory. To run the sample query programs successfully a vocabulary must have been loaded.

1. Enter:

```
cd {LEXBIG_DIRECTORY}/examples
```

2. Execute one of sample programs. .bat for windows or .sh for Linux.

- FindConceptNameForCode.bat
- FindPropsandAssocForCode.bat
- FindRelatedCodes
- FindTreeforCodeAndAssoc

The following figure shows sample program output for finding properties and associations for a given code.

```

C:\Program Files\LexGrid\LexBIG\2.3.0rc2\examples>FindPropsAndAssocForCode.bat T005
# | Local Name | Version | Tag |
--|---|---|---|
1 | NCI_Thesaurus | 05.09.bvt | SAMPLE |
2 | UMLS_SemNet | 3.2 | SAMPLE |
--|---|---|---|

NOTE: >> indicates column value exceeds the available width.
Enter the number of the Coding scheme to use, then <Enter> :
2
Pointed at by ...
affects
    T039/Physiologic Function
    T041/Mental Process
    T038/Biologic Function
    T040/Organism Function
    T191/Neoplastic Process
    T190/Anatomical Abnormality
    T049/Cell or Molecular Dysfunction
    T050/Experimental Model of Disease
    T048/Mental or Behavioral Dysfunction
    T019/Congenital Abnormality
    T047/Disease or Syndrome
    T020/Acquired Abnormality
    T046/Pathologic Function
    T045/Genetic Function
    T044/Molecular Function
    T043/Cell Function
    T042/Organ or Tissue Function
interacts_with
    T005/Virus
    T004/Fungus
    T003/Alga
    T002/Plant
part_of
    T028/Gene or Genome
    T026/Cell Component
    T190/Anatomical Abnormality
  
```

Installing NCI Vocabularies

NCI Thesaurus Vocabulary

This section describes the steps to download and install a full version of the NCI Thesaurus for the LexEVS Service.

1. Using a web or ftp client go to URL: <http://ftp1.nci.nih.gov/pub/cacore/EVS/>

Name	Size	Type	Modified
archive		File Folder	2/6/2006 1:24 PM
caBIG_lexGrid		File Folder	6/22/2005 12:00 AM
CDR		File Folder	6/14/2005 12:00 AM
fda		File Folder	10/18/2005 5:46 PM
protege		File Folder	10/19/2005 4:33 PM
ThesaurusSemantics		File Folder	2/6/2006 1:17 PM
ThesaurusTermsofUse_files		File Folder	9/23/2003 12:00 AM
Filtered_pipe_out.zip	0.97 MB	Compressed (zipped)...	8/26/2005 12:00 AM
Filtered_pipe_out_0601c.txt	64.0 KB	Text Document	2/6/2006 10:04 AM
Filtered_pipe_out_12f.txt	78.0 KB	Text Document	1/11/2006 12:18 PM
full_pipe_out.zip	2.25 MB	Compressed (zipped)...	7/1/2005 12:00 AM
full_pipe_out_0601c.txt	210 KB	Text Document	2/6/2006 10:04 AM
full_pipe_out_12f.txt	471 KB	Text Document	1/11/2006 12:18 PM
Metathesaurus_P051117.zip	625 MB	Compressed (zipped)...	1/27/2006 5:39 PM
mmsys.a.prop	19.0 KB	PROP File	5/25/2004 12:00 AM
MMSYS.jar	6.70 KB	Executable Jar File	5/25/2004 12:00 AM
mmsys.prop.sav	19.0 KB	SAV File	5/25/2004 12:00 AM
NCI_RRF_Addendum.pdf	130 KB	Adobe Acrobat Doc...	7/22/2004 12:00 AM
NCI_THESAURUS_license.txt	6.33 KB	Text Document	10/21/2003 12:00 AM
onlylog.dtd	7.18 KB	DTD File	1/4/2006 3:44 PM
ReadMe.txt	4.89 KB	Text Document	2/6/2006 10:38 AM
ReadMe_history.txt	3.70 KB	Text Document	2/6/2006 10:38 AM
Thesaurus_05.11f.FLAT.zip	3.04 MB	Compressed (zipped)...	1/4/2006 3:43 PM
Thesaurus_05.11f.OWL.zip	6.76 MB	Compressed (zipped)...	1/4/2006 3:43 PM
Thesaurus_05.11f.XML.zip	6.98 MB	Compressed (zipped)...	1/4/2006 3:43 PM
Thesaurus_05.12f.FLAT.zip	3.12 MB	Compressed (zipped)...	2/6/2006 10:18 AM
Thesaurus_05.12f.OWL.zip	6.86 MB	Compressed (zipped)...	2/6/2006 10:18 AM
Thesaurus_05.12f.XML.zip	7.10 MB	Compressed (zipped)...	2/6/2006 10:18 AM
ThesaurusTermsofUse.htm	10.8 KB	HTML Document	10/21/2003 12:00 AM
ThesaurusTermsofUse.pdf	82.2 KB	Adobe Acrobat Doc...	3/29/2005 12:00 AM

2. Select the version of NCI Thesaurus OWL you wish to download. Save the file to a directory on your machine.
3. Extract the OWL file from the zip download and save in a directory on your machine. This directory will be referred to as NCI_THESAURUS_DIRECTORY
4. Using the LexEVS utilities load the NCI Thesaurus

```
cd {LexBIG_DIRECTORY}/admin
```

For Windows installation use the following command:

```
LoadNCIThesOWL.bat -nf -in "file:/// {NCI_THESAURUS_DIRECTORY}/Thesaurus_05.12f.owl"
```

For Linux installation use the following command:

```
LoadNCIThesOWL.sh -nf -in "file:/// {NCI_THESAURUS_DIRECTORY}/Thesaurus_05.12f.owl"
```



Note

This step will require about three hours on a Pentium 3.0 Ghz machine. The total time to load NCI Thesaurus will vary depending on machine, memory, and disk speed.

The following example shows output from load of NCI Thesaurus 05.12f:

```
...
[LexBIG] Processing TOP Node... Retired_Kind
[LexBIG] Clearing target of NCI_Thesaurus...
[LexBIG] Writing NCI_Thesaurus to target...
```

```

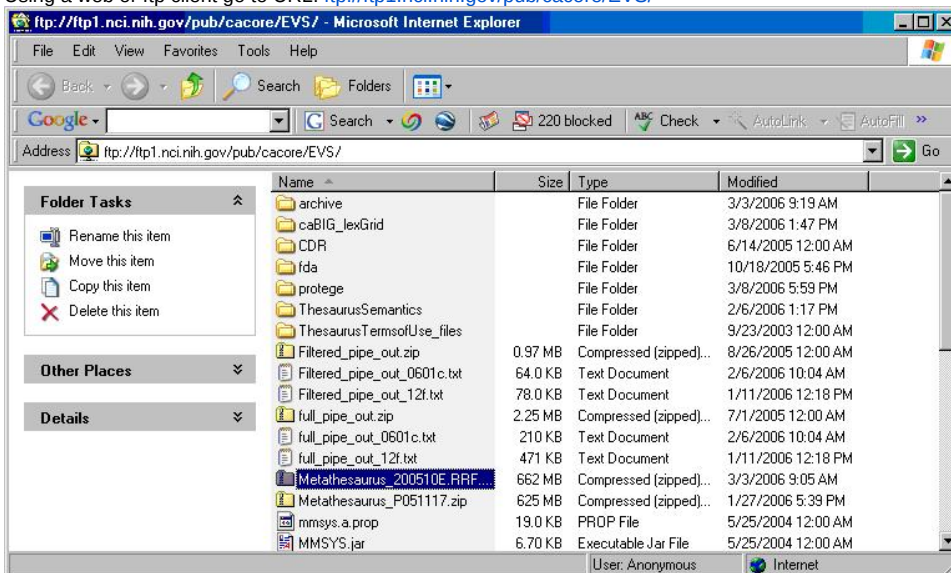
[LexBIG] Finished loading DB - loading transitive expansion table
[LexBIG] ComputeTransitive - Processing Anatomic_Structure_Has_Location
[LexBIG] ComputeTransitive - Processing Anatomic_Structure_is_Physical_Part_of
[LexBIG] ComputeTransitive - Processing Biological_Process_Has_Initiator_Process
[LexBIG] ComputeTransitive - Processing Biological_Process_Has_Result_Biological_Process
[LexBIG] ComputeTransitive - Processing Biological_Process_Is_Part_of_Process
[LexBIG] ComputeTransitive - Processing Conceptual_Part_Of
[LexBIG] ComputeTransitive - Processing Disease_Excludes_Finding
[LexBIG] ComputeTransitive - Processing Disease_Has_Associated_Disease
[LexBIG] ComputeTransitive - Processing Disease_Has_Finding
[LexBIG] ComputeTransitive - Processing Disease_May_Have_Associated_Disease
[LexBIG] ComputeTransitive - Processing Disease_May_Have_Finding
[LexBIG] ComputeTransitive - Processing Gene_Product_Has_Biochemical_Function
[LexBIG] ComputeTransitive - Processing Gene_Product_Has_Chemical_Classification
[LexBIG] ComputeTransitive - Processing Gene_Product_is_Physical_Part_of
[LexBIG] ComputeTransitive - Processing hasSubtype
[LexBIG] Finished building transitive expansion - building index
[LexBIG] Getting a results from sql (a page if using mysql)
[LexBIG] Indexed 0 concepts.
[LexBIG] Indexed 5000 concepts.
[LexBIG] Indexed 10000 concepts.
[LexBIG] Indexed 15000 concepts.
[LexBIG] Indexed 20000 concepts.
[LexBIG] Indexed 25000 concepts.
[LexBIG] Indexed 30000 concepts.
[LexBIG] Indexed 35000 concepts.
[LexBIG] Indexed 40000 concepts.
[LexBIG] Indexed 45000 concepts.
[LexBIG] Indexed 46000 concepts.
[LexBIG] Getting a results from sql (a page if using mysql)
[LexBIG] Closing Indexes Mon, 27 Feb 2006 01:44:22
[LexBIG] Finished indexing

```

NCI Metathesaurus Vocabulary

This section describes the steps to download and install a full version of the NCI Metathesaurus for the LexEVS Service.

1. Using a web or ftp client go to URL: <ftp://ftp1.nci.nih.gov/pub/cacore/EVS/>



2. Select the version of NCI Metathesaurus RRF you wish to download. Save the file to a directory on your machine.
3. Extract the RRF files from the zip download and save in a directory on your machine. This directory will be referred to as NCI_METATHESAURUS_DIRECTORY.



Note

RELEASE_INFO.RRF is required to be present for the load utility to work.

4. Using the LexEVS utilities load the NCI Thesaurus

```
cd {LexBIG_DIRECTORY}/admin
```

For Windows installation use the following command:

```
LoadNCIMeta.bat -nf -in "file:/// {NCI_METATHESAURUS_DIRECTORY}/"
```

For Linux installation use the following command:

```
LoadNCIMeta.sh -nf -in "file:/// {NCI_THESAURUS_DIRECTORY}/"
```



Note

NCI Metathesaurus contains many individual vocabularies and requires several hours to load and index. This step requires about 15 hours on a Pentium 3.0 Ghz machine with 7200rpm disk. The total time to load NCI MetaThesaurus will vary depending on machine, memory, and disk speed.

NCI History

This section describes the steps to download and install a history file for NCI Thesaurus.

1. Using a web or ftp client go to URL: <ftp://ftp1.nci.nih.gov/pub/cacore/EVS/>
2. Select the version of NCI History you wish to download. Save the file to a directory on your machine. Select the VersionFile download to the same directory as the history file.
3. Extract the History files from the zip download and save in a directory on your machine. This directory will be referred to as NCI_HISTORY_DIRECTORY
4. Using the LexEVS utilities load the NCI Thesaurus:

```
cd {LexBIG_DIRECTORY}/admin_*
```

For Windows installation use the following command:

```
LoadNCIHistory.bat -nf -in "file:/// {NCI_HISTORY_DIRECTORY}" -vf "file:/// {NCI_HISTORY_DIRECTORY} /VersionFile"
```

For Linux installation use the following command:

```
LoadNCIHistory.sh -nf -in "file:/// {NCI_HISTORY_DIRECTORY}" -vf "file:/// {NCI_HISTORY_DIRECTORY} /VersionFile"
```



Note

If a 'releaseId' occurs twice in the file, the last occurrence will be stored. If LexEVS already knows about a releaseId (from a previous history load), the information is updated to match what is provided in the file.

This file has to be provided to the load API on every load because you will need to maintain it in the future as each new release is made. We have created this file that should be valid as of today from the information that we found in the archive folder on your ftp server. You can find this file in the 'resources' directory of the LexEVS install.

Deactivating and Removing a Vocabulary

This section describes the steps to deactivate a coding scheme and remove coding scheme from LexEVS Service.

1. Change directory to LexEVS administration directory. Enter:

```
cd {LEXBIG_DIRECTORY}/admin_*
```

2. Use the DeactivateScheme utility to prevent access to coding scheme. Once a coding scheme is deactivated, client programs will not be able to access the content for the specific coding scheme and version. Example:

```
DeactivateScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.12f"
```

3. Use RemoveScheme utility to remove coding scheme from LexEVS service and database. Example:

```
RemoveScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.12f"
```

Tagging a Vocabulary

This section describes the steps to tag a coding scheme to be used via LexEVS API.

1. Change directory to LexEVS administration directory and enter:

```
cd {LEXBIG_DIRECTORY}/admin_*
```

2. Use the TagScheme to tag a coding system and version with a local tag name (e.g. PRODUCTION). This tag name can be used via LexEVS API for query restriction. Example:

```
TagScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.12f" -t "PRODUCTION"
```