

# 2 - Administering LexEVS 6.0 with the Command Line

## Contents of this Page

- [Vocabulary Administration Overview](#)
  - [Coding Scheme Metadata Access and Administration](#)
  - [Lucene Index Administration](#)
  - [Misc Database Administration Scripts](#)
  - [LexEVS Export Scripts](#)
  - [LexEVS Loader Scripts](#)
  - [Special Batch Loading Functions](#)
  - [Scheme and Metadata Removal](#)
  - [Pick List and Value Set Load Administration](#)
  - [Expert Admin Scripts \(Focused on Mass Loads of Value Sets\)](#)
  - [LexEVS Validation Test Script](#)
- [Command Line Scripts and Wrappers Overview](#)
  - [Displaying Current Coding Schemes](#)
  - [Using Optional Parameters](#)
  - [Setting Java Virtual Machine Options](#)
- [Installing Sample Vocabularies](#)
- [Running the sample query programs](#)
- [Installing NCI Vocabularies](#)
  - [NCI Thesaurus Vocabulary](#)
  - [NCI Metathesaurus Vocabulary](#)
    - [Loading the Metathesaurus](#)
    - [Resuming Loads](#)
  - [NCI History](#)
- [Deactivating and Removing a Vocabulary](#)
- [Tagging a Vocabulary](#)
- [IndexManagement](#)
  - [Remove Index](#)
  - [Rebuild Index](#)
- [Value Set and Pick List Loading](#)

## LexEVS Administration Links

- [Admin Guide Main Page](#)
  - [Admin with LexEVS GUI](#)
  - [Admin with Command Line](#)
  - [Management and Admin API](#)
  - [Advanced Vocab Admin](#)
- [LexEVS 6.0 Main Page](#)
- [LexEVS Current Release](#)

## Vocabulary Administration Overview

A set of administrative shell scripts are provided to manage the LexEVS Service. These scripts are provided for Windows (.bat) and Linux (.sh) operating systems and call wrapper classes for LexEVS administration API classes. These scripts are located in the {LEXBIG\_DIRECTORY}/admin and {LEXBIG\_DIRECTORY}/test directory. A full description of the options with an example is provided for each command line script.

If you are loading a large vocabulary such as SNOMED or the NCI Metathesaurus, then we recommend the use of these scripts over the GUI since adjusting system memory requirements can be accomplished by changing the load script with a text editor.



## Coding Scheme Metadata Access and Administration

Shell Script	Use and Function
--------------	------------------

ActivateScheme	<p>Activates a coding scheme based on unique URN and version. Options:</p> <ul style="list-style-type: none"> <li>• -u,-urn &lt;urn&gt; URN uniquely identifying the code system.</li> <li>• -v,-version &lt;versionId&gt; Version identifier.</li> <li>• -f,-force Force activation (no confirmation).</li> </ul> <p>Example:</p> <pre>ActivateScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.09e"</pre>
DeactivateScheme	<p>Deactivates a coding scheme based on unique URN and version. Options:</p> <ul style="list-style-type: none"> <li>• -u,-urn &lt;urn&gt; URN uniquely identifying the code system.</li> <li>• -v,-version &lt;versionId&gt; Version identifier.</li> <li>• -d,-date &lt;yyyy-MM-dd,HH:mm:ss&gt; Date and time for deactivation to take effect; immediate if not specified.</li> <li>• -f,-force Force deactivation (no confirmation).</li> </ul> <p>Example:</p> <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>
ListExtensions	<p>List registered extensions to the LexEVS runtime environment. Options:</p> <ul style="list-style-type: none"> <li>• -a,-all List all extensions (default, override by specifying other options).</li> <li>• -i,-index List index extensions.</li> <li>• -m,-match List match algorithm extensions.</li> <li>• -s,-sort List sort algorithm extensions.</li> <li>• -g,-generic List generic extensions.</li> </ul> <p>Example:</p> <pre>ListExtensions -a</pre>
ListSchemes	<p>List all currently registered vocabularies. Options:</p> <ul style="list-style-type: none"> <li>• -b,-brief List only coding scheme name, version, urn, and tags (default).</li> <li>• -f,-full List full detail for each scheme.</li> </ul> <p>Example: ListSchemes</p>
TagScheme	<p>Associates a tag ID (e.g. 'PRODUCTION' or 'TEST') with a coding scheme URN and version. Options:</p> <ul style="list-style-type: none"> <li>• -u,-urn &lt;urn&gt; URN uniquely identifying the code system.</li> <li>• -v,-version &lt;versionId&gt; Version identifier.</li> <li>• -t,-tag The tag ID (e.g. 'PRODUCTION' or 'TEST') to assign.</li> </ul> <p>Example:</p> <pre>TagScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v 05.09e" -t "TEST"</pre>

## Lucene Index Administration

Shell Script	Use and Function
--------------	------------------

CleanUpLuceneIndex	<p>Clean up orphaned indexes. Options:</p> <ul style="list-style-type: none"> <li>• -h, -help, Prints usage information</li> <li>• -r, -reindex, Reindex any missing indices.</li> </ul> <div>  <b>Note</b>            Lucene Clean Up can only be executed in the default Single Index Mode.         </div> <p>Example:</p> <pre>CleanUPLuceneIndex -r</pre>
OptimizeLuceneIndex	<p>Optimizes the Common Lucene Index. Options:</p> <ul style="list-style-type: none"> <li>• -h, -help, Prints usage information</li> </ul>
RebuildIndex	<p>Rebuilds indexes associated with the specified coding scheme. Options:</p> <ul style="list-style-type: none"> <li>• -u,-urn &lt;urn&gt; URN uniquely identifying the code system.</li> <li>• -v, -version &lt;versionId&gt; Version identifier.</li> <li>• -i,-index &lt;name&gt; Name of the index extension to rebuild (if absent, rebuilds all built-in indices and named extensions).</li> <li>• -f,-force Force clear (no confirmation).</li> </ul> <p>Example:</p> <pre>RebuildIndex -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.09e" -i "myindex"</pre>
RemoveIndex	<p>Clears an optional named index associated with the specified coding scheme.</p> <div>  <b>Note</b>            Built-in indices required by the LexEVS runtime cannot be removed.         </div> <p>Options:</p> <ul style="list-style-type: none"> <li>• -u,-urn &lt;urn&gt; URN uniquely identifying the code system.</li> <li>• -v,-version &lt;versionId&gt; Version identifier.</li> <li>• -i,-index &lt;name&gt; Name of the index extension to clear.</li> <li>• -f,-force Force clear (no confirmation).</li> </ul> <p>Example:</p> <pre>RemoveIndex -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.09e"</pre>




## Misc Database Administration Scripts

Shell Script	Use and Function
--------------	------------------

ExportDDLScripts	<p>Exports the database create/drop scripts. Options:</p> <ul style="list-style-type: none"> <li>• -db, -databaseType, required=true, usage="Target database type.")</li> <li>• -p, -prefix Prefix to append to all tables</li> <li>• -o, -out Output directory, a required option</li> <li>• -f, -force Force output directory creation.</li> </ul> <p>Example:</p> <pre>ExportDDLScripts -p lb -o "file:///path/to/dir" -f</pre>
PasswordEncryptor	<p>Encrypts the given password. Argument:</p> <ul style="list-style-type: none"> <li>• -p, -password password for encryption.</li> </ul> <p>Example:</p> <pre>PasswordEncryptor -p lexgrid</pre>

## LexEVS Export Scripts


Shell Script	Use and Function
ExportLgXML	<p>Exports content from the repository to a file in the LexGrid canonical XML format. Usage: ExportLgXML [-xc] [-an null] [-xall] -v null -u name -out uri [-xa] [-f]</p> <ul style="list-style-type: none"> <li>• -a, -associationsName Export associations with this name. Only valid with export type 'xa'</li> <li>• -f, -force If specified, allows the destination file to be overwritten if present.</li> <li>• -out, -output &lt;uri&gt; URI or path of the directory to contain the resulting XML file. The file name will be automatically derived from the coding scheme name.</li> <li>• -u, -urn &lt;name&gt; URN or local name of the coding scheme to export.</li> <li>• -v, -version The assigned tag/label or absolute version identifier of the coding scheme.</li> <li>• -xa, -exportAssociations Type of export: export only associations.</li> <li>• -xall, -exportAll Type of export: export all content. Default behavior.</li> <li>• -xc, -exportConcepts Type of export: export only concepts.</li> </ul> <p>Example:</p> <pre>ExportLgXML -out "file:///path/to/dir" -u "NCI Thesaurus" -v "05.06e" -f</pre>

ExportOBO	<p>Exports content from the repository to a file in the Open Biomedical Ontologies (OBO) format.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -out,-output &lt;uri&gt; URI or path of the directory to contain the resulting OBO file. The file name will be automatically derived from the coding scheme name.</li> <li>• -u,-urn &lt;name&gt; URN or local name of the coding scheme to export.</li> <li>• -v,-version &lt;id&gt; The assigned tag/label or absolute version identifier of the coding scheme.</li> <li>• -nf,-noFail If specified, indicates that processing should not stop for recoverable errors</li> <li>• -f,-force If specified, allows the destination file to be overwritten if present.</li> </ul> <div data-bbox="315 365 1477 499"> <p> <b>Note</b></p> <p>If the coding scheme and version values are unspecified, a list of available coding schemes will be presented for user selection.</p> </div> <div data-bbox="315 520 1477 630"> <p> <b>Note</b></p> <p>Available for Linux based OS only.</p> </div> <p>Examples:</p> <div data-bbox="315 739 1477 814"> <pre>ExportOBO -out "file:///path/to/dir" -nf -f</pre> </div> <div data-bbox="315 835 1477 911"> <pre>ExportOBO -out "file:///path/to/dir" -u "FBbt" -v "PRODUCTION" -nf -f</pre> </div>
ExportOwlRdf	<p>Exports content from the repository to a file in OWL format.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -out,-output &lt;uri&gt; URI or path of the directory to contain the resulting OWL file. The file name will be automatically derived from the coding scheme name.</li> <li>• -u,-urn &lt;name&gt; URN or local name of the coding scheme to export.</li> <li>• -v,-version &lt;id&gt; The assigned tag/label or absolute version identifier of the coding scheme.</li> <li>• -f,-force If specified, allows the destination file to be overwritten if present.</li> </ul> <div data-bbox="315 1155 1477 1264"> <p> <b>Note</b></p> <p>If the URN and version values are unspecified, a list of available coding schemes will be presented for user selection.</p> </div> <p>Examples:</p> <div data-bbox="315 1373 1477 1449"> <pre>ExportOwlRdf -out "file:///path/to/dir" -f</pre> </div> <div data-bbox="315 1470 1477 1545"> <pre>ExportOwlRdf -out "file:///path/to/dir" -u "sample" -v "1.0" -f</pre> </div>

## LexEVS Loader Scripts


Shell Script	Use and Function
--------------	------------------

LoadLgXML	<p>Loads a vocabulary file, provided in LexGrid canonical xml format.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in,-input &lt;uri&gt; URI specifying location of the source file.</li> <li>• -v, -validate &lt;level&gt; Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: <ul style="list-style-type: none"> <li>◦ 0 = Verify document is well-formed</li> <li>◦ 1 = Verify document is valid</li> </ul> </li> <li>• -nf,-noFail If specified, indicates that processing should not stop for recoverable errors.</li> <li>• -a,-activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated.</li> <li>• -t, -tag &lt;tagID&gt; An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign.</li> </ul> <p>Load Example:</p> <pre>LoadLgXML -in "file:///path/to/file.xml" -nf -a</pre> <p>Validation Example:</p> <pre>LoadLgXML -in "file:///path/to/file.xml" -v 0</pre>
LoadNCIHistory	<p>Imports NCI History data to the LexEVS repository.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in,-input &lt;uri&gt; URI specifying location of the history file</li> <li>• -vf,-versionFile &lt;uri&gt; URI specifying location of the file containing version identifiers for the history to be loaded.</li> <li>• -v, -validate &lt;level&gt; Perform validation of the candidate resource without loading data. If specified, the '-nf' and '-r' options are ignored. Supported levels of validation include: <ul style="list-style-type: none"> <li>◦ 0 = Verify top 10 lines are correct format</li> <li>◦ 1 = Verify correct format for the entire file</li> </ul> </li> <li>• -nf,-noFail If specified, indicates that processing should not stop for recoverable errors</li> <li>• -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.</li> </ul> <p>Load Example:</p> <pre>LoadNCIHistory -nf -in "file:///path/to/history.file" -vf "[file:///path/to/version.file]"</pre> <p>Validation Example:</p> <pre>LoadNCIHistory -in "file:///path/to/history.file" -v 0</pre> <p>Versions File Format Information:</p> <pre>releaseDate   isLatest   releaseAgency   releaseId   releaseOrder   entityDescription</pre> <p>Sample record:</p> <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>

LoadOBO	<p>Loads a file specified in the Open Biomedical Ontologies (OBO) format. Options:</p> <ul style="list-style-type: none"> <li>• -in,-input &lt;uri&gt; URI or path specifying location of the source file</li> <li>• -v,-validate &lt;int&gt; Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: <ul style="list-style-type: none"> <li>◦ 0 = Verify document is valid</li> </ul> </li> <li>• -nf,-noFail If specified, indicates that processing should not stop for recoverable errors</li> <li>• -a, -activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated</li> <li>• -t, -tag &lt;id&gt; An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign.</li> </ul> <p>Examples:</p> <pre>LoadOBO -in "file:///path/to/file.obo" -nf -a</pre> <pre>LoadOBO -in "file:///path/to/file.obo" -v 0</pre>
LoadOWL	<div data-bbox="269 684 1463 793">  <b>Note</b>  Load of the NCI Thesaurus should be performed via the LoadNCIThesOWL counterpart, since it will allow more precise handling of NCI semantics. </div> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in,-input &lt;uri&gt; URI or path specifying location of the source file</li> <li>• -v,-validate &lt;int&gt; Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: <ul style="list-style-type: none"> <li>◦ 0 = Verify document is well-formed</li> <li>◦ 1 = Verify document is valid</li> </ul> </li> <li>• -nf,-noFail If specified, indicates that processing should not stop for recoverable errors</li> <li>• -a, -activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated</li> <li>• -t, -tag &lt;id&gt; An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign.</li> </ul> <p>Example:</p> <pre>LoadOWL -in "file:///path/to/somefile.owl" -nf -a</pre> <pre>LoadOWL -in "file:///path/to/somefile.owl" -v 0</pre>
LoadRadLexProtegeFrames	<p>URI or path specifying location of the pprj file. Imports from a RadLex xml file to a LexBIG repository. Requires that the pprj file be configured with reference to a RadLex xml file as follows:</p> <pre>([radlex_ProjectKB_Instance_66] of String (name "source_file_name") (string_value "radlex.xml"))" );</pre> <p>Example:</p> <pre>java org.LexGrid.LexBIG.admin.LoadRadLexProtegeFrames</pre> <ul style="list-style-type: none"> <li>• -in,-input &lt;uri&gt; URI or path specifying location of the source file</li> <li>• -v,-validate &lt;int&gt; Perform validation of the candidate resource without loading data. If specified, the '-a' and '-t' options are ignored. Supported levels of validation include: <ul style="list-style-type: none"> <li>◦ 0 = Verify document is well-formed</li> <li>◦ 1 = Verify document is valid</li> </ul> </li> <li>• -a, -activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated</li> <li>• -t, -tag &lt;id&gt; An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign.</li> </ul> <p>Example:</p> <pre>java -Xmx800m -cp lgRuntime.jar -Djava.ext.dirs=../runtime/sqlDrivers org.LexGrid.LexBIG.admin.LoadRadLexProtegeFrames -in "file:///path/to/RadLex.pprj" -a -or- org.LexGrid.LexBIG.admin.LoadRadLexProtegeFrames -in "file:///path/to/RadLex.pprj" -v 0</pre>

LoadUMLS SSemnet	<p>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:</p> <ul style="list-style-type: none"> <li>• LICENSE.txt (text from distribution terms and conditions)</li> <li>• SRFIL.txt (File Description)</li> <li>• SRFIL.txt (Field Description)</li> <li>• SRDEF.txt (Basic information about the Semantic Types and Relations)</li> <li>• SRSTR.txt (Structure of the Network)</li> <li>• SRSTRE1.txt (Fully inherited set of Relations (UIs))</li> <li>• SRSTRE2.txt (Fully inherited set of Relations (Names))</li> <li>• SU.txt (Unit Record)</li> </ul> <p>These files can be downloaded from the NLM web site at <a href="http://semanticnetwork.nlm.nih.gov/Download/index.html">http://semanticnetwork.nlm.nih.gov/Download/index.html</a>.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in, -input &lt;uri&gt; URI or path of the directory containing the NLM files</li> <li>• -v, -validate &lt;int&gt; Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: <ul style="list-style-type: none"> <li>◦ 0 = Verify the existence of each required file</li> </ul> </li> <li>• -nf, -noFail If specified, indicates that processing should not stop for recoverable errors</li> <li>• -a, -activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated.</li> <li>• -t, -tag &lt;id&gt; An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign.</li> <li>• -il -InheritanceLevel &lt;int&gt; 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.</li> </ul> <p>Example:</p> <pre>LoadUMLSSemnet -in "file:///path/to/directory/" -nf --a --il 1</pre> <pre>LoadUMLSSemnet -in "file:///path/to/directory/" -v 0</pre>
LoadFMA	<p>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 <a href="http://sig.biustr.washington.edu/projects/fm/">http://sig.biustr.washington.edu/projects/fm/</a> upon registration.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in, -input &lt;uri&gt; URI or path specifying location of the source file</li> <li>• -v, -validate &lt;int&gt; Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: <ul style="list-style-type: none"> <li>◦ 0 = Verify document is well-formed</li> <li>◦ 1 = Verify document is valid</li> </ul> </li> <li>• -nf, -noFail If specified, indicates that processing should not stop for recoverable errors</li> <li>• -a, -activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated</li> <li>• -t, -tag &lt;id&gt; An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign.</li> </ul> <p>Examples:</p> <pre>LoadFMA -in "file:///path/to/FMA.pprj" -nf -a</pre> <pre>LoadFMA -in "file:///path/to/FMA.pprj" -v 0</pre>
LoadHL7 RIM	<p>Converts an HL7 RIM MS Access database to a LexGrid database</p> <ul style="list-style-type: none"> <li>• -in, -input &lt;uri&gt; URI or path specifying location of the source file</li> <li>• -mf, -manifest &lt;uri&gt; URI or path specifying location of the manifest file</li> <li>• -lp, -load preferences &lt;uri&gt; URI or path specifying location of the load preferences file</li> <li>• -v, -validate &lt;int&gt; Perform validation of the candidate resource without loading data. If specified, the '-nf', '-a' and '-t' options are ignored. Supported levels of validation include: <ul style="list-style-type: none"> <li>◦ 0 = Verify document is valid</li> </ul> </li> <li>• -nf, -noFail If specified, indicates that processing should not stop for recoverable errors</li> <li>• -a, -activate ActivateScheme on successful load; if unspecified the vocabulary is loaded but not activated</li> <li>• -t, -tag &lt;id&gt; An optional tag ID (e.g. 'PRODUCTION' or 'TEST') to assign.</li> </ul> <p>Example:</p> <pre>LoadHL7RIM -in "file:///path/to/file.mdb" -nf -a</pre> <pre>LoadHL7RIM -in "file:///path/to/file.mdb" -v 0</pre>



LoadMeta Data	<p>Loads optional XML-based metadata to be associated with an existing coding scheme.</p> <ul style="list-style-type: none"> <li>• -u,-urn &lt;name&gt; URN uniquely identifying the code system.</li> <li>• -v,-version &lt;id&gt; Version identifier.</li> <li>• -in,-input &lt;uri&gt; URI or path specifying location of the XML file.</li> <li>• -v,-validate &lt;int&gt; 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</li> <li>• -o, -overwrite If specified, existing metadata for the code system will be erased. Otherwise, new metadata will be appended to existing metadata (if present).</li> <li>• -f,-force Force overwrite (no confirmation).</li> <li>• -nf,-noFail If specified, indicates that processing should not stop for recoverable errors</li> </ul> <div>  <b>Note</b>            If the URN and version values are unspecified, a list of available coding schemes will be presented for user selection.         </div> <p>Examples:</p> <pre>LoadMetadata -in "file:///path/to/file.xml" -nf -o</pre> <pre>LoadMetadata -in "file:///path/to/file.xml"</pre>
LoadMrMap	<p>Loads mappings file(s), provided in UMLS RRF format. Specifically MRMAP.RRF and MRSAT.RRF.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -inMap,-input &lt;uri&gt; URI or path specifying location of the MRMAP source file.</li> <li>• -inSat,-input &lt;uri&gt; URI or path specifying location of the MRSAT source file.</li> </ul> <p>Example:</p> <pre>LoadMrMap -inMap "file:///path/to/MRMAP.RRF" -inSat "file:///path/to/MRSAT.RRF"</pre>

## Special Batch Loading Functions

Shell Script	Use and Function
LoadMetaBatch	<p>Loads the NCI MetaThesaurus, provided as a collection of RRF files, using a batch loading strategy allowing a faster, more memory efficient load to occur.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in,-input &lt;uri&gt; URI or path of the directory containing the NLM files</li> </ul> <p>Example:</p> <pre>LoadMetaBatch -in "file:///path/to/directory/"</pre>
LoadUMLBatch	<p>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 <a href="http://www.nlm.nih.gov/research/umls/metaa1.html">http://www.nlm.nih.gov/research/umls/metaa1.html</a>.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in,-input &lt;uri&gt; URI or path of the directory containing the NLM files</li> <li>• -s,-source vocabularies to load.</li> </ul> <p>Example:</p> <pre>LoadUMLSBatch -in "file:///path/to/directory/" -s "PSY"</pre>

ResumeMetaBatch	<p>Resume a UMLS load. Loads will usually be restartable if they fail due to an error. The loader will keep all loaded content and restart at the point of failure.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in,-input &lt;uri&gt; URI or path of the directory containing the NLM files</li> <li>• -s,-source vocabulary to resume.</li> <li>• -uri,-uri of vocabulary to resume.</li> <li>• -version,-version of vocabulary to resume.</li> </ul> <p>Example:</p> <pre>ResumeMetaBatch -in "file:///path/to/directory/" -s "PSY" -uri "urn:123.4" -version "2.0"</pre>
ResumeUmlsBatch	<p>Resume a UMLS load. Loads will usually be restartable if they fail due to an error. The loader will keep all loaded content and restart at the point of failure.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in,-input &lt;uri&gt; URI or path of the directory containing the NLM files</li> <li>• -s,-source vocabulary to resume.</li> <li>• -uri,-uri of vocabulary to resume.</li> <li>• -version,-version of vocabulary to resume.</li> </ul> <p>Example:</p> <pre>ResumeUmlsBatch.sh -in "file:///path/to/directory/" -s "PSY" -uri "urn:123.4" -version "2.0"</pre>

## Scheme and Metadata Removal

Shell Script	Use and Function
RemoveScheme	<p>Removes a coding scheme based on unique URN and version.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -u,-urn &lt;urn&gt; URN uniquely identifying the code system.</li> <li>• -v,-version &lt;versionId&gt; Version identifier.</li> <li>• -f,-force Force deactivation and removal without confirmation.</li> </ul> <p>Example:</p> <pre>RemoveScheme -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.09e"</pre>
RemoveMetadata	<p>Clears optionally loaded metadata associated with the specified coding scheme.</p> <p>Options</p> <ul style="list-style-type: none"> <li>• -u,-urn &lt;urn&gt; URN uniquely identifying the code system.</li> <li>• -v,-version &lt;id&gt; Version identifier.</li> <li>• -f,-force Force clear (no confirmation).</li> </ul> <div>  <b>Note</b>        If the URN and version values are unspecified, a list of available coding schemes will be presented for user selection.     </div> <p>Examples:</p> <pre>RemoveMetadata</pre> <pre>RemoveMetadata -u "urn:oid:2.16.840.1.113883.3.26.1.1" -v "05.09e"</pre>

## Pick List and Value Set Load Administration

Shell Script	Use and Function
LoadPickListDefinition	<p>Loads Pick List Definition content, provided in LexGrid canonical xml format.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in, -input &lt;uri&gt; URI or path specifying location of the source file.</li> <li>• -v, -validate &lt;int&gt; Perform validation of the candidate resource without loading data. Supported levels of validation include: <ul style="list-style-type: none"> <li>◦ 0 = Verify document is well-formed</li> <li>◦ 1 = Verify document is valid</li> </ul> </li> </ul> <p>Example:</p> <pre>LoadPickListDefinition -in "file:///path/to/file.xml"</pre>
LoadAllDefinitionsToResolvedValueSet	<p>Loads Value Set Definition content, provided in LexGrid canonical xml format.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -in, -input &lt;uri&gt; URI or path specifying location of the source file.</li> <li>• -v, -validate &lt;int&gt; Perform validation of the candidate resource without loading data. Supported levels of validation include: <ul style="list-style-type: none"> <li>◦ 0 = Verify document is well-formed</li> <li>◦ 1 = Verify document is valid</li> </ul> </li> </ul> <p>Example:</p> <pre>LoadValueSetDefinition -in "file:///path/to/file.xml"</pre>

## Expert Admin Scripts (Focused on Mass Loads of Value Sets)




These Scripts have a narrow focus on mass loads and removals of value sets. They can be very destructive.

Shell Script	Use and Function
LoadAllValueSetDefinitionsInDirectory	<p>Loads All LexGrid XML formatted Value Set Definitions in a given directory.</p> <p>Highly recommended that the folder be cleaned of all non value set files and folders</p> <p>Options:</p> <ul style="list-style-type: none"> <li>◦ -in, Path to directory</li> </ul> <p>Example:</p> <pre>LoadAllValueSetDefinitionsInDirectory -in /path/to/directory</pre>
LoadAllDefinitionsToResolvedValueSet	<p>Loads All Value Set Definitions for a given uri as resolved coding scheme. All compiled resolved value sets should be removed to insure resolution updates are correct with the current version of the source terminology.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• -u, URI uniquely identifying the coding scheme to resolve from</li> <li>• defaults to <a href="http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#">http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#</a></li> </ul> <p>Example:</p> <pre>LoadAllDefinitionsToResolvedValueSet -uri "http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#"</pre>

RemoveAllValueSetDefinitions	<p>WARNING: Removes all value set definitions without discrimination</p> <p>Options:</p> <p>None: Takes no input, removes all value set definitions in the system. Cannot be reversed.</p> <p>Example:</p> <pre>./RemoveAllValueSetDefinitions.sh</pre>
RemoveAllResolvedValueSets	<p>WARNING: REMOVES ALL RESOLVED VALUE SETS</p> <p>Options: None, removes all resolved value sets from the system. Not reversible.</p> <p>Example:</p> <pre>./RemoveAllResolvedValueSets.sh</pre>

## LexEVS Validation Test Script

Shell Script	Use and Function
TestRunner	<p>Located in {LEXBIG_DIRECTORY}/test. Runs the test suite by invoking the Ant launcher.</p> <div>  <b>Note</b>  The LexEVS runtime and database environments must still be configured prior to invoking the test suite. Running any option other than -v will cause a large set of junits to be run </div> <p>Usage: TestRunner</p> <ul style="list-style-type: none"> <li>• -b, -brief Run the LexBIG test suite and produce a text report with overall statistics and details for failed tests only.</li> <li>• -f, -full Run the LexBIG test suite and produce an itemized list of all tests with indication of success/failure.</li> <li>• -x, -xml Run the LexBIG test suite and produce a report with extensive information for each test case in xml format.</li> <li>• -h, -html Run the LexBIG test suite and produce a report suitable for view in a standard web browser: this is the default if no other option is specified.</li> <li>• -v, -verify Basic verification that LexEVS is configured properly and basic systems are functioning.</li> </ul> <p>Examples:</p> <pre>TestRunner -v</pre> <p>Run the full test suite of jUnits and save the report to an html file. (This can take quite some time.)</p> <pre>TestRunner -f -h</pre> <p>Run a verification of the LexEVS installation. A quick way to insure your system is up and running.</p> <pre>TestRunner -v</pre>

## Command Line Scripts and Wrappers Overview

LexEVS provides a set of shell scripts and code wrappers that provide command line users standard options and automatic printouts of current coding schemes when performing administrative functions using the command line.

## Displaying Current Coding Schemes

When you have coding Schemes Loaded to LexEVS many command line script calls will bring up a printout of a table of the loaded coding schemes to be used as a selective menu for other functions.

CPT	2010	urn:oid:2.16.840.1.113883.6.12	null
Thesaurus.owl	05.09.bvt	http://ncicb.nci.nih.gov/xml/owl/>>	null
NCI_Thesaurus	10.10a	http://ncicb.nci.nih.gov/xml/owl/>>	
MDR:MDR12_1_TO_CST>>	200909	urn:oid:CL413321.MDR.CST	null
MDR:MDR12_1_TO_ICD>>	200909	urn:oid:CL413320.MDR.ICD9CM	null
Automobiles_extens>>	1.0-exte>>	urn:oid:11.11.0.1.1-extension	null
GO	October2>>	urn:lsid:bioontology.org:GO	PRODUCTION
NPO	1.0_Jan_>>	http://purl.bioontology.org/ontolo>>	null
SNOMEDCT_2010_01_3>>	20100131	urn:oid:C2733618.SNOMEDCT.ICD9CM	null

NOTE: >> indicates column value exceeds the available width.  
Specify the '-f' option for additional detail.

[m029206@bmidev4 admin]\$

## Using Optional Parameters

A standard parameter interface allows users to customize command line calls to LexEVS. These options control things such as loader synchronicity, supplemental file locations, and fault conditions. Shell scripts contain comments on usage for each function wrappers options.

```
# Loads an OWL file. You can provide a manifest file to configure coding scheme
# meta data.
#
# Options:
# -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,--loaderPrefs<uri> URI or path specifying location of the loader preference file
# -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: LoadOWL -in "file:///path/to/somefile.owl" -a
#
java -Xmx3000m -XX:MaxPermSize=256M -cp "../runtime/lbPatch.jar:../runtime/lbRuntime.jar"

org.LexGrid.LexBIG.admin.LoadOWL $@
```

## Setting Java Virtual Machine Options

Heap size:

Memory management for heap size is a particular issue when loading larger Terminologies such as SNOMED or the NCI Thesaurus. Users need to be prepared to increase heap size if loads crash with heap size errors.

Example:

```
-Xmx:4000m
```

Permanent Generation:

LexEVS now depends on some Permanent Generation memory management during runtime.

Typically set as follows:

```
-XX:MaxPermSize:256m
```

## Installing Sample Vocabularies

This LexEVS installation provides a sample vocabulary, Automobiles.xml, 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 an example vocabulary, run the LoadSampleData script, LoadSampleData.bat for Windows; LoadSampleData.sh for Linux.
3. To load other example vocabularies, run the appropriate script for any sample vocabulary in {LEXBIG\_DIRECTORY}/test/resources/testData.



### Note

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

```
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 be loaded.

1. Enter

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

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

```

c:\ Command Prompt
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

```
LoadOWL.bat -in "file:/// {NCI_THESAURUS_DIRECTORY} /Thesaurus_10.10d.owl"
```

- For Linux installation use the following command

```
LoadOWL.sh -in "file:/// {NCI_THESAURUS_DIRECTORY} /Thesaurus_10.10d.owl"
```



#### Note

The NCI Thesaurus has grown large enough that it can no longer be loaded on many typical desktop machines. We recommend a 64-bit operating system running on a multiprocessor computer with a minimum of 4g of memory. Server class Linux machines are the typical target for these loads. The time to load NCI Thesaurus will vary depending on machine, memory, and disk speed. Expect a couple of hours for a higher end machine.

The following code sample shows the example 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

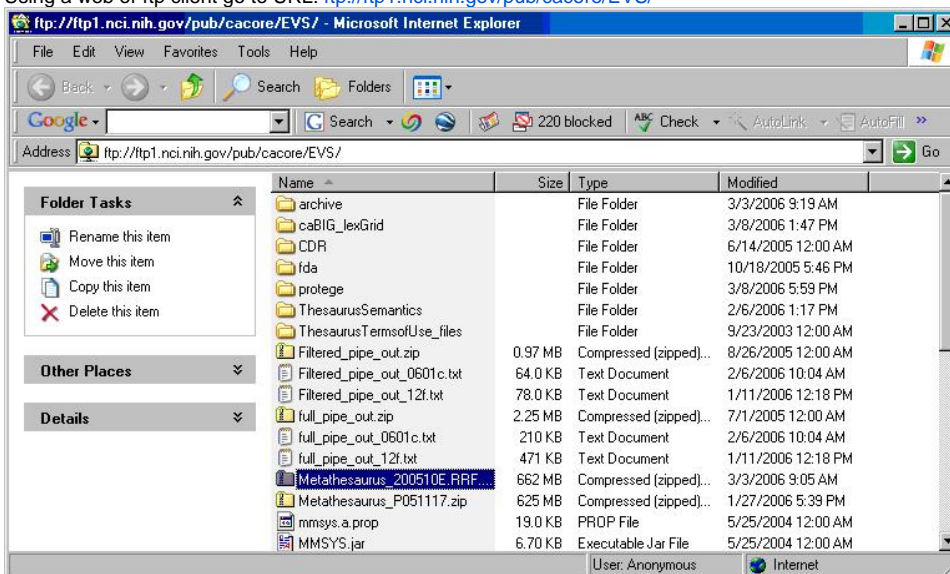
```

## NCI Metathesaurus Vocabulary

### Loading the Metathesaurus

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. 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

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

For Linux installation use the following command

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



#### Note

NCI Metathesaurus contains many individual vocabularies some of which are large vocabularies in and of themselves. It requires many hours to load and index. It can require 36 hours on a multiprocessor machine with 6g plus memory. The total time to load NCI MetaThesaurus will vary depending on machine, memory, and disk speed. Because this loader uses a batch loading strategy it is less dependent on memory, but some users will see 3 or 4 day load times with average multiprocessor processing power.

## Resuming Loads

Since this loader is resource hungry we provide the option to restart should you find your resource settings to be inadequate. Resuming loads which have crashed or been interrupted by server problems is possible using the ResumeBatchLoad script set.

1. Using the LexEVS utilities load the NCI Thesaurus

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

- For Windows installation use the following command

```
ResumeMetaBatch.bat -in "file:/// {NCI_METATHESAURUS_DIRECTORY} /" -s "NCI Metathesaurus" -uri "urn:oid:2.16.840.1.113883.3.26.1.2" -version "200601"
```

- For Linux installation use the following command

```
ResumeMetaBatch.sh -in "file:/// {NCI_THESAURUS_DIRECTORY} /" -s "NCI Metathesaurus" -uri "urn:oid:2.16.840.1.113883.3.26.1.2" -version "200601"
```

## 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://ftpl.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 'releaseld' occurs twice in the file, the last occurrence will be stored. If LexEVS already knows about a releaseld (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 "10.10d"
```

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 "10.10d"
```

## 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. 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 "10.10d" -t "PRODUCTION"
```

## IndexManagement

LexEVS indexes can be manually updated, removed and optimized by users. This is a useful function particularly when metadata, authoring and post processing may have taken place.

### Remove Index

1. Change directory to LexEVS administration directory. Enter:

```
cd {{LEXBIG_DIRECTORY}}/admin
```

2. Use RemoveIndex script to remove an index for a particular coding scheme.

Example:

Windows:

```
RemoveIndex.bat
```

Linux:

```
RemoveIndex.sh
```

3. Interactive command: Enter the number of the Coding Scheme to use, then <Enter>. User responds with column number for appropriate coding scheme.

- Interactive command: CLEAR? ('Y' to confirm, any other key to cancel). User responds with 'Y'
- Command Line example:

```
C:\WINDOWS\system32\cmd.exe

C:\Lex6\admin>RemoveIndex.bat
log4j:ERROR Could not find value for key log4j.appender.debug
log4j:ERROR Could not instantiate appender named "debug"
LB_WARN_LOGGER WARN - 2010-11-29 12:44:30.441 - [LOG ID 10] No user defined Cache
SLF4J: This version of SLF4J requires log4j version 1.2.12 or later. See also htt
net.sf.ehcache.config.ConfigurationFactory WARN - 2010-11-29 12:44:32.379 - No co

# | Local Name | Version | Tag |
1 | Automobiles | 1.0 | SAMPLE |
2 | amino-acid | 1.2 | null |

NOTE: >> indicates column value exceeds the available width.
Enter the number of the Coding scheme to use, then <Enter> :
2
CLEAR? <'Y' to confirm, any other key to cancel>
y
[LB] [Mon Nov 29 12:44:40 CST 2010] Request complete
C:\Lex6\admin>_
```

#### Note

Concepts cannot be resolved without an index. Rebuilding the index is required.

## Rebuild Index

- Change directory to LexEVS administration directory. Enter

```
cd {LEXBIG_DIRECTORY}/admin
```

- Use RebuildIndex script to rebuild an index for a particular coding scheme.

Example:

Windows:

```
RebuildIndex.bat
```

Linux:

```
RebuildIndex.sh
```

- Interactive command: Enter the number of the Coding Scheme to use, then <Enter>.

User responds with column number for appropriate coding scheme.

- Interactive command: REBUILD INDEX FOR URI? <uri displayed here> ('Y' to confirm, any other key to cancel). User responds with 'Y'

## 2. Command Line example:

```
C:\WINDOWS\system32\cmd.exe

C:\Lex6\admin>RebuildIndex.bat
log4j:ERROR Could not find value for key log4j.appender.debug
log4j:ERROR Could not instantiate appender named "debug".
LB_WARN_LOGGER WARN - 2010-11-29 12:57:16.747 - [LOG_ID 10] No user defined Cache
SLF4J: This version of SLF4J requires log4j version 1.2.12 or later. See also htt
net.sf.ehcache.config.ConfigurationFactory WARN - 2010-11-29 12:57:18.294 - No co

# | Local Name | Version | Tag |
--|---|---|---|
1 | Automobiles | 1.0 | SAMPLE |
2 | amino-acid | 1.2 | null |

NOTE: >> indicates column value exceeds the available width.
Enter the number of the Coding scheme to use, then <Enter> :
2
REBUILD INDEX FOR URI: http://www.co-ode.org/ontologies/amino-acid/2006/05/18/ani
y
[LB] [Mon Nov 29 12:58:33 CST 2010] Recreation of index extension 'URI: http://ww
[LB] [Mon Nov 29 12:58:33 CST 2010] beginning process
LB_WARN_LOGGER WARN - 2010-11-29 12:58:34.047 - [LOG_ID 12] Using default cache f
[LB] [Mon Nov 29 12:58:34 CST 2010] Indexed: 0 Entities.
[LB] [Mon Nov 29 12:58:35 CST 2010] Finished processes

C:\Lex6\admin>
```



### Note

It may be useful and performance enhancing to run OptimizeLuceneIndex script after removing and rebuilding.

## Value Set and Pick List Loading

[Value Set and Pick List Scripts](#)