

# 3 - Administering LexEVS 6.1 with the Command Line

## Contents of this Page

- [LexEVS 6.1 Loader Scripts](#)


## 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](#)

LexEVS 6.1 has all the administrative capabilities of LexEVS 6.0 as well as the following loader functions:

## LexEVS 6.1 Loader Scripts

Shell Script	Use and Function
LoadMedDRA	<p>Loads a file specified in the Medical Dictionary for Regulatory Activities (MedDRA) 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>• -mf,--manifest &lt;uri&gt; URI or path specifying location of the manifest 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:0 = Verify document is valid</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>Load Example:</p> <div>LoadMedDRA -in "file:///path/to/file.asc" -a</div> <p>Validation Example:</p> <div>LoadMedDRA -in "file:///path/to/file.asc" -v 0</div>

LoadMIFVocabulary	<p>Loads from an the HL7 Vocabulary mif file.</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>• -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>Load Example:</p> <pre>LoadMIFVocabulary -in "file:///path/to/file.xml" -a</pre> <div>  <b>Note</b>  This replaces the LoadHL7RIM script and function </div>
LoadOWL2	<p>Loads an OWL file. You can provide a manifest file to configure coding scheme# meta data.</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>• -mf,--manifest &lt;uri&gt; URI or path specifying location of the manifest file</li> <li>• -lp,--loaderPrefs&lt;uri&gt; URI or path specifying location of the loader preference 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>Load Example:</p> <pre>LoadOWL2 -in "file:///path/to/somefile.owl" -a</pre> <pre>LoadOWL2 -in "file:///path/to/somefile.owl" -v 0</pre>
LoadResolvedValueSetDefinition	<p>Loads Value Set Definition content, provided in LexGrid canonical xml format.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>-u, The valueset definition URI to use</li> <li>-l, The list of coding schemes to revolve against. The format is codingschemeName::version</li> <li>-csVersionTag The tag to use for resolving coding scheme</li> </ul> <p>Load Example:</p> <pre>LoadResolvedValueSetDefinition -u \"Automobiles:valuesetDefinitionURI\" -l \"Automobiles::version1, GM::version2\" -csVersionTag \"production\" \" \"</pre>

: