1 - LexEVS 6.x Administration Using the GUI Tool

Contents of this Page

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

Launching the GUI Administration Tool

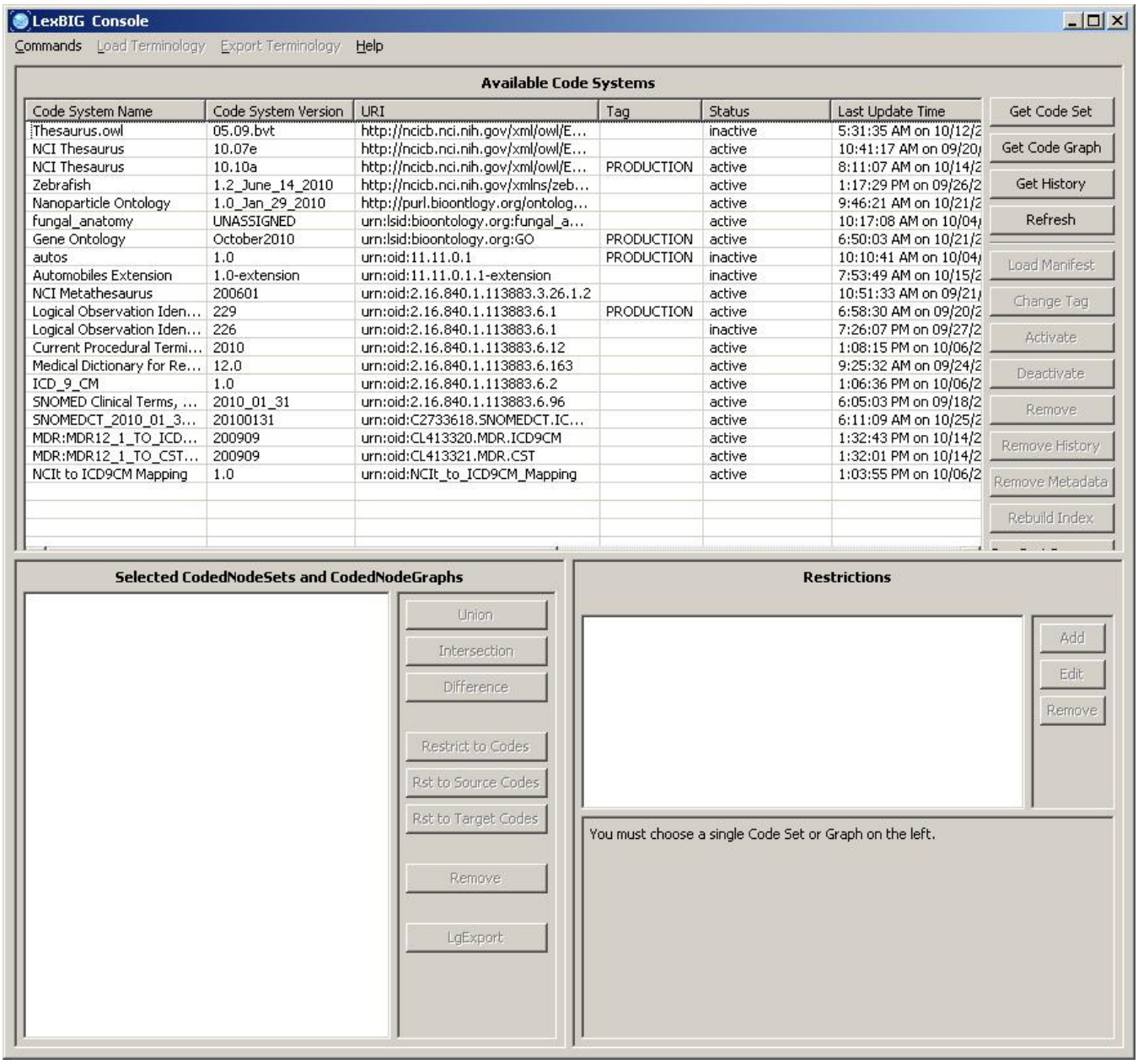
If you choose to install the LexEVS GUI when you installed LexEVS, you will have a `'gui'` folder inside of your LexEVS base installation. You should have the following programs in the `'gui'` folder:

```
Linux_64-lbGUI.sh
Linux-lbGUI.sh
Windows-lbGUI.bat
Windows-lbGUI-browser.bat
```

We provide two Windows shell script versions which allow a choice between the full-fledged interface for loading, managing, and end-use, or the interface for end-use only.

This shell script provides an example by which any shell script can pass an argument option "-d" into the java command launching the LexEVS GUI application, restricting the end user to browsing only, and allowing no loading or management of terminologies.

Launch the GUI by executing the appropriate script for your platform. You will be presented with an application that looks like this:



This application will let you perform most administrative functions that are available in the LexEVS API. To enable the administrative functions, first go to the 'Commands' menu, and then click on the 'Enable Admin Options' submenu. This will enable all of the commands that can make changes to the LexEVS environment.

This guide covers only the administrative commands. Refer to the *LexEVS Programmer's Guide* for instructions on using the rest of the LexEVS GUI.

Administrative Menu Commands

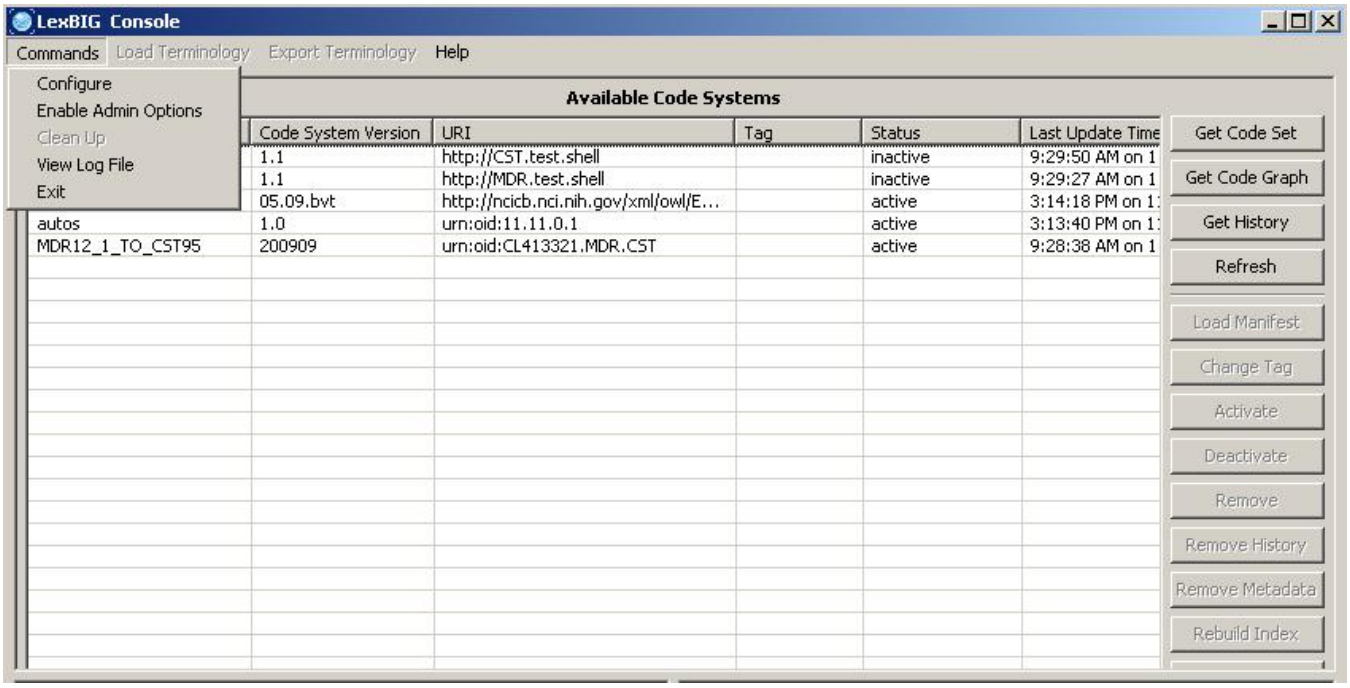
Commands Menu

Menu Item	Menu Action
Configure	This menu option will bring up a dialog which will show you all of the options from the current `lbconfig.props` file. You can make changes to individual variable here - but these changes will only affect the GUI - they will not be written back out to the `lbconfig.props` file. You can also choose which `lbconfig.props` file that you want to use.
Enable Admin Options	This option enables or disables all of the GUI features which are considered administrative options.

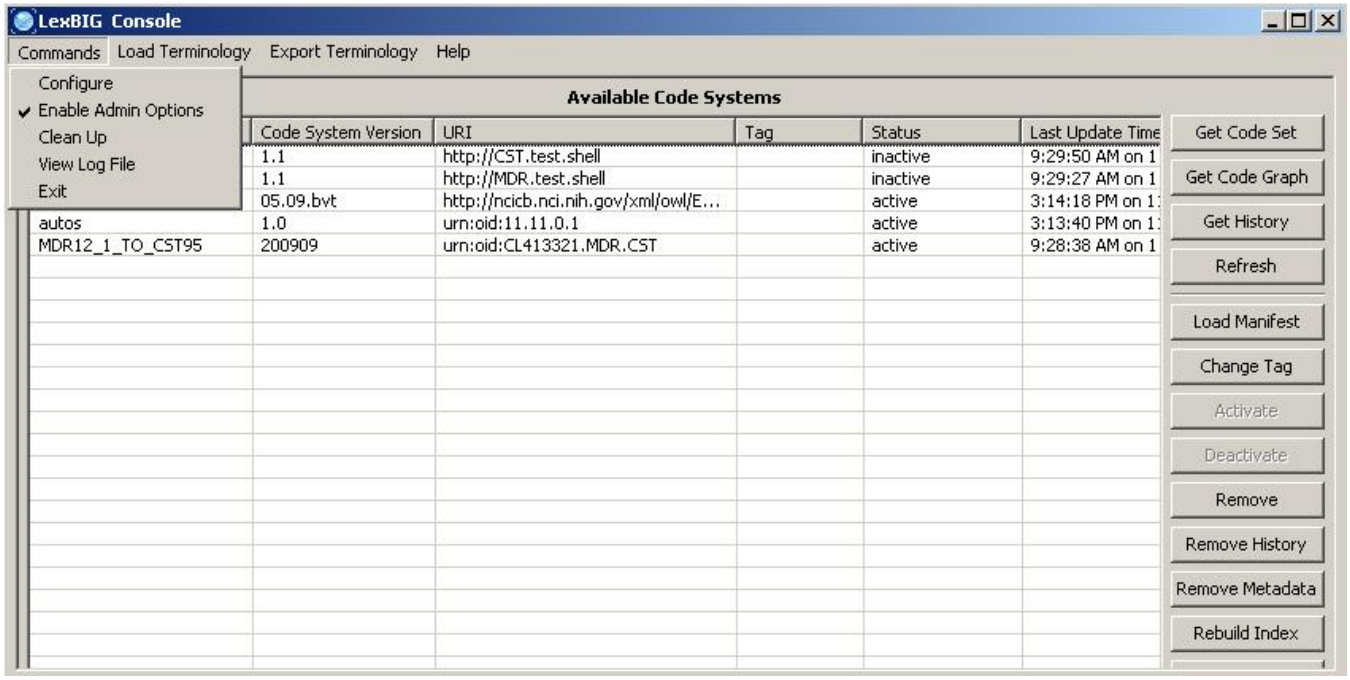
Clean Up	This command will run the clean up orphaned resources tool. It will give you a listing of any resources that are orphaned in the LexEVS environment, and give you the option to remove them.
View Log File	This will show you the all of the logs messages that have occurred during the LexEVS GUI session. The log file viewer also has choices to let you customize the types of messages that are logged.
Exit	Close the application.

Enabling Administration Options

The drop down menu labeled "Command" provides general administrative access.



Selecting 'Enable Admin Options' provides administrative access to all commands

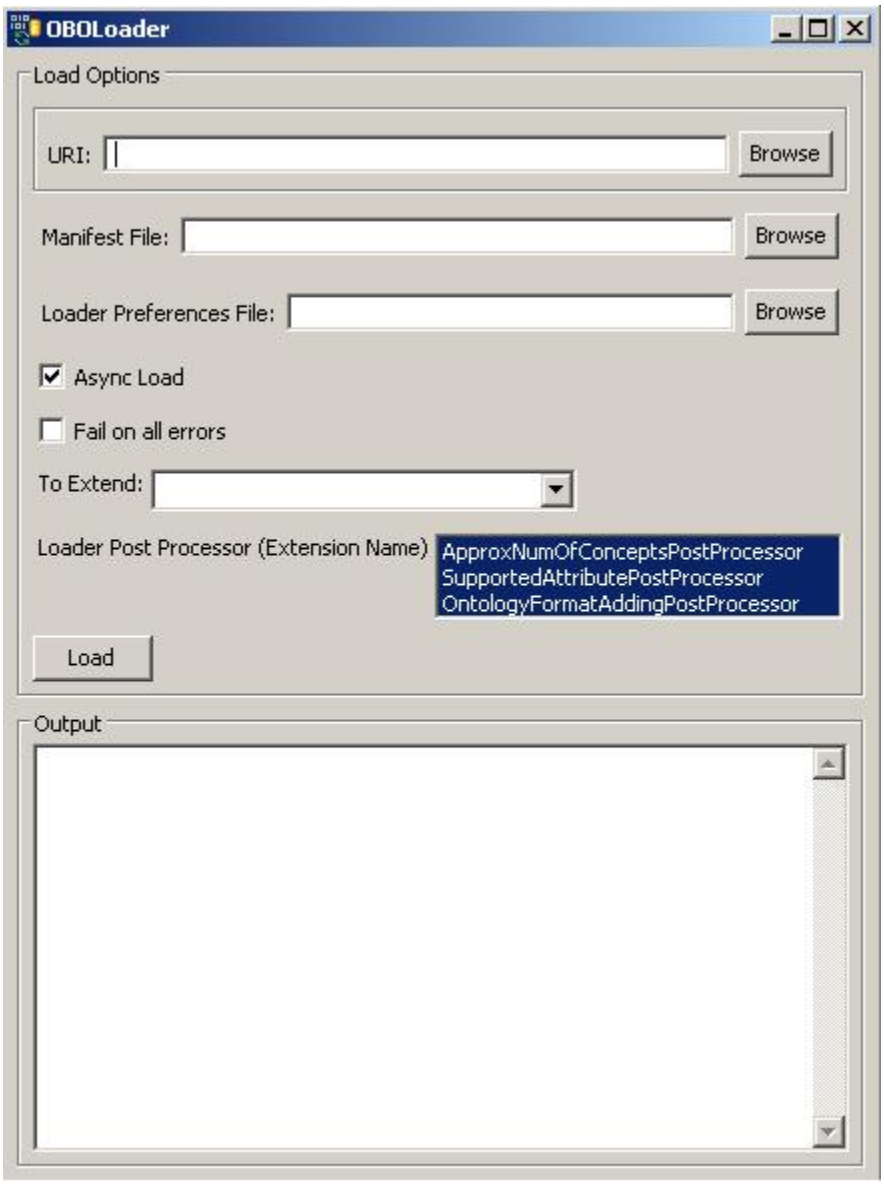


Load Terminology Menu

Loader Type in Menu	Menu Action
Various Loaders (Standard version)	This menu option launches terminology loader menus which have the same interface
History Loaders	This menu option launches history loader menus which have the same interface
Text Loader	This menu option launches a text loader menu with a unique interface
MrMap Loader	This menu option launches a an RRF Loader for the MRMAP, MRSAT file set loader menu with a unique interface

Standard Loaders

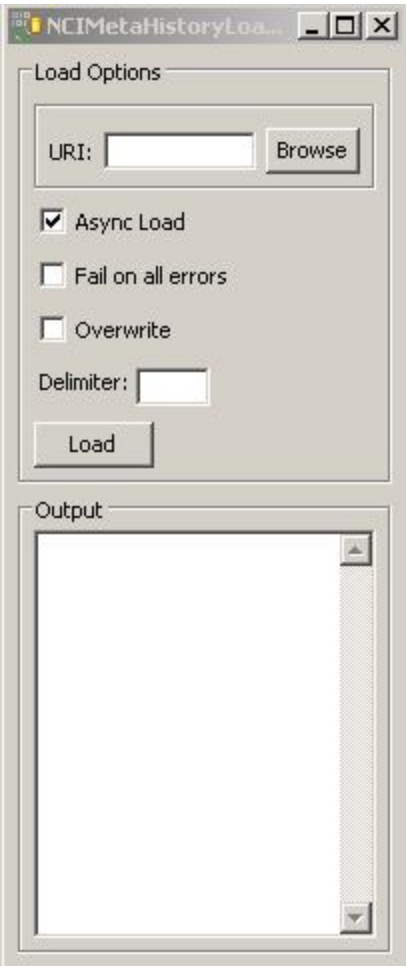
Most loaders have the same interface and option set:



- *URI*: The path to the file or folder containing source material for the load.
- *Manifest file*: The path to a manifest file that will change or add values to the coding schemes metadata
- *Loader Preferences file*: The path to a preferences file that will allow the user to decide how some source formats, such as OWL, will be loaded.
- *Async load*: By default the loader loads as an asynchronous thread to other LexEVS activities allowing the JVM to continue if the loader fails.
- *Fail on all errors*: Load will fail on any source error if this is checked.
- *To Extend*: You may be loading this coding scheme as an extension to another. The drop down menu allows the user to make a choice between loaded coding schemes.
- *Load Post Processor (Extension name)*: The user may choose selected post processor algorithms to run on the coding scheme immediately after it loads.

History Loaders

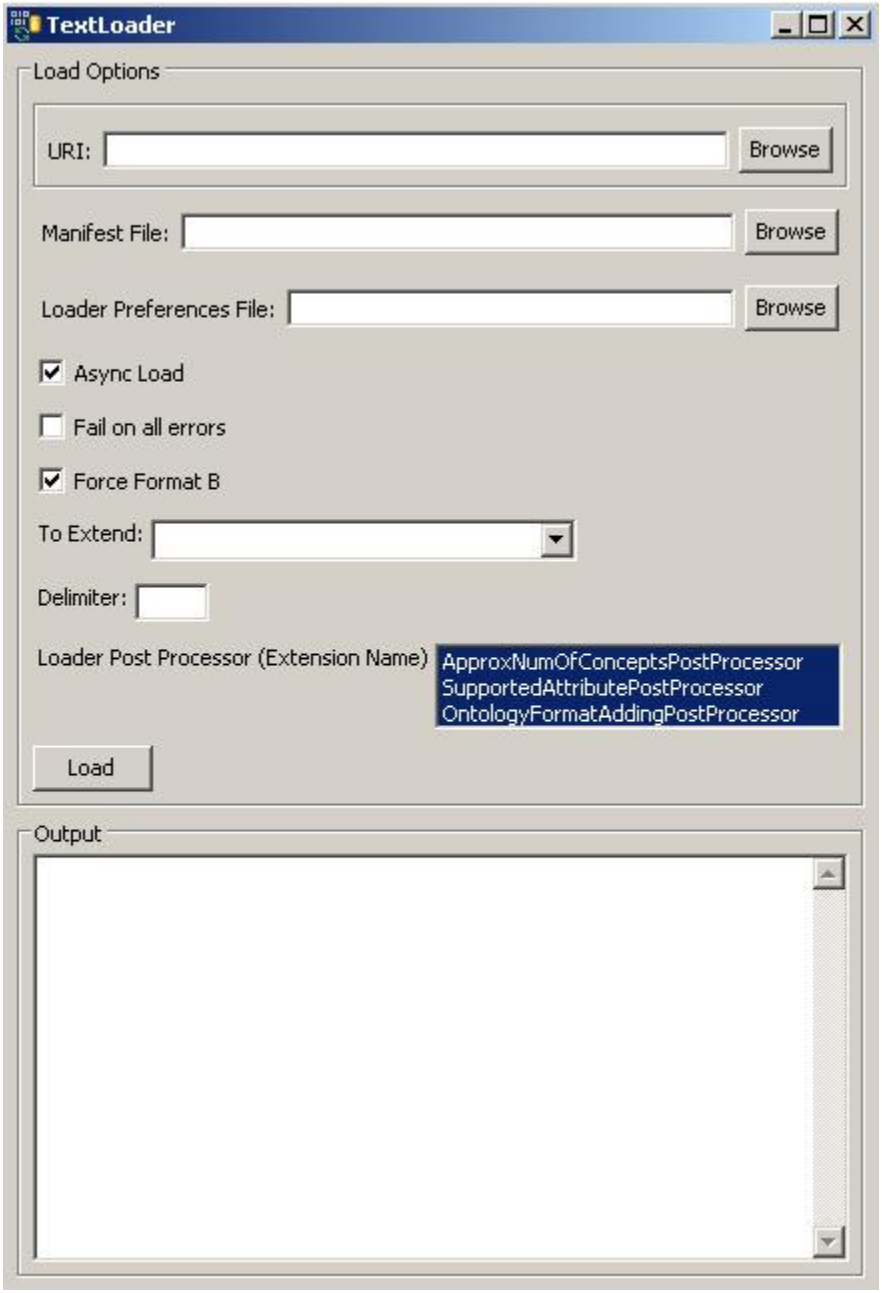
Change history can be loaded for selected source types:



- *URI:* The path to the file or folder containing source material for the load.
- *Async load:* By default the loader loads as an asynchronous thread to other LexEVS activities allowing the JVM to continue if the loader fails.
- *Fail on all errors:* Load will fail on any source error if this is checked.
- *Overwrite:* Overwrite current history for this source.
- *Delimiter:* Designate the delimiter token for this source file.

Text Loader

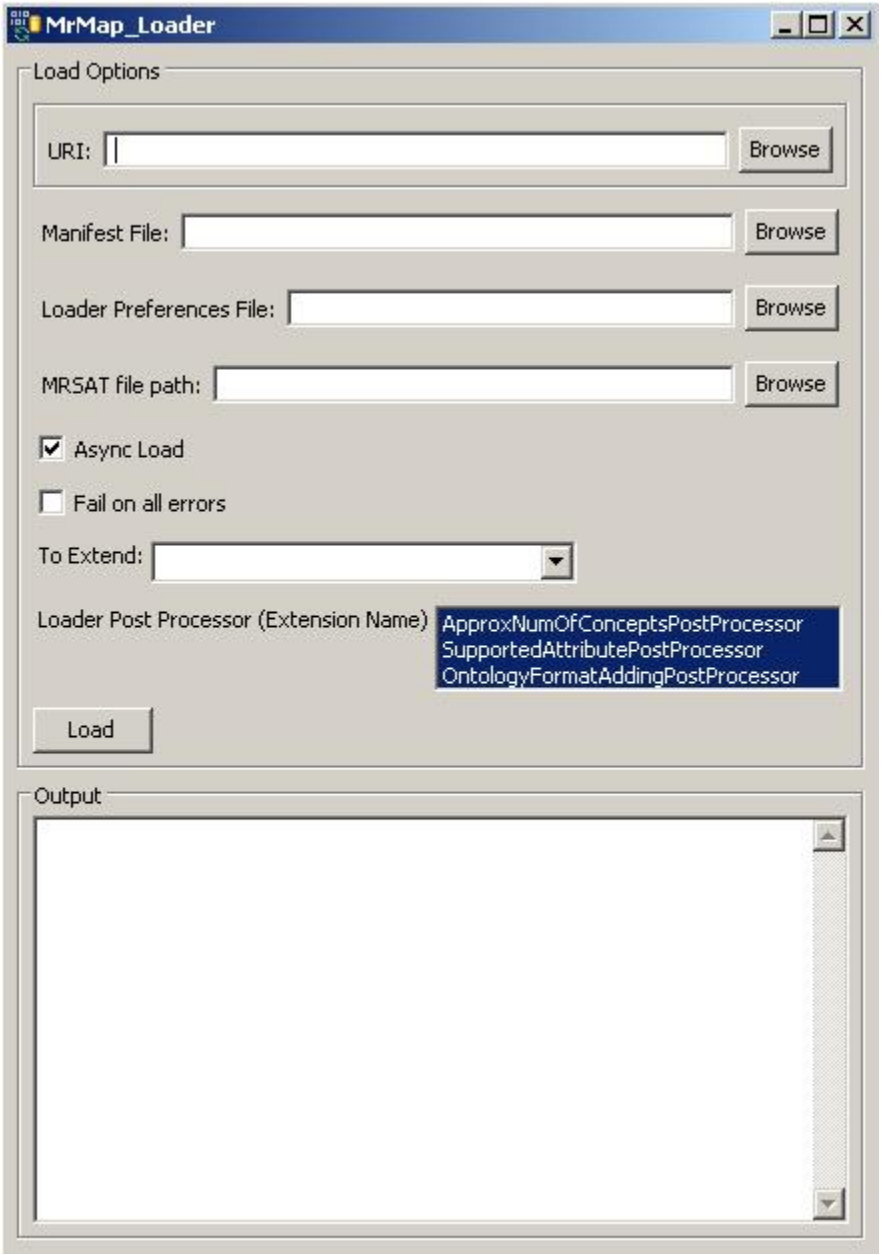
Loader for text files formatted as LexGrid text



- *URI:* The path to the file or folder containing source material for the load.
- *Async load:* By default the loader loads as an asynchronous thread to other LexEVS activities allowing the JVM to continue if the loader fails.
- *Fail on all errors:* Load will fail on any source error if this is checked.
- *Force Format B:* Force the converter to read a doubles file (name/description) as a triples file (code/name/description) So it reads codes and names instead of names and descriptions.
- *Overwrite:* Overwrite current history for this source.
- *Delimiter:* Designate the delimiter token for this source file.
- *To Extend:* You may be loading this coding scheme as an extension to another. The drop down menu allows the user to make a choice between loaded coding schemes.
- *Load Post Processor (Extension name):* The user may choose selected post processor algorithms to run on the coding scheme immediately after it loads.

MrMapLoader

Loader class for MrMap and MrSat RRF files resulting in a mapping coding scheme or schemes depending on the content of these files. Some standard Loader Options are not available for the MrMap Loader since applying metadata or preferences to multiple mapping schemes would not be appropriate.

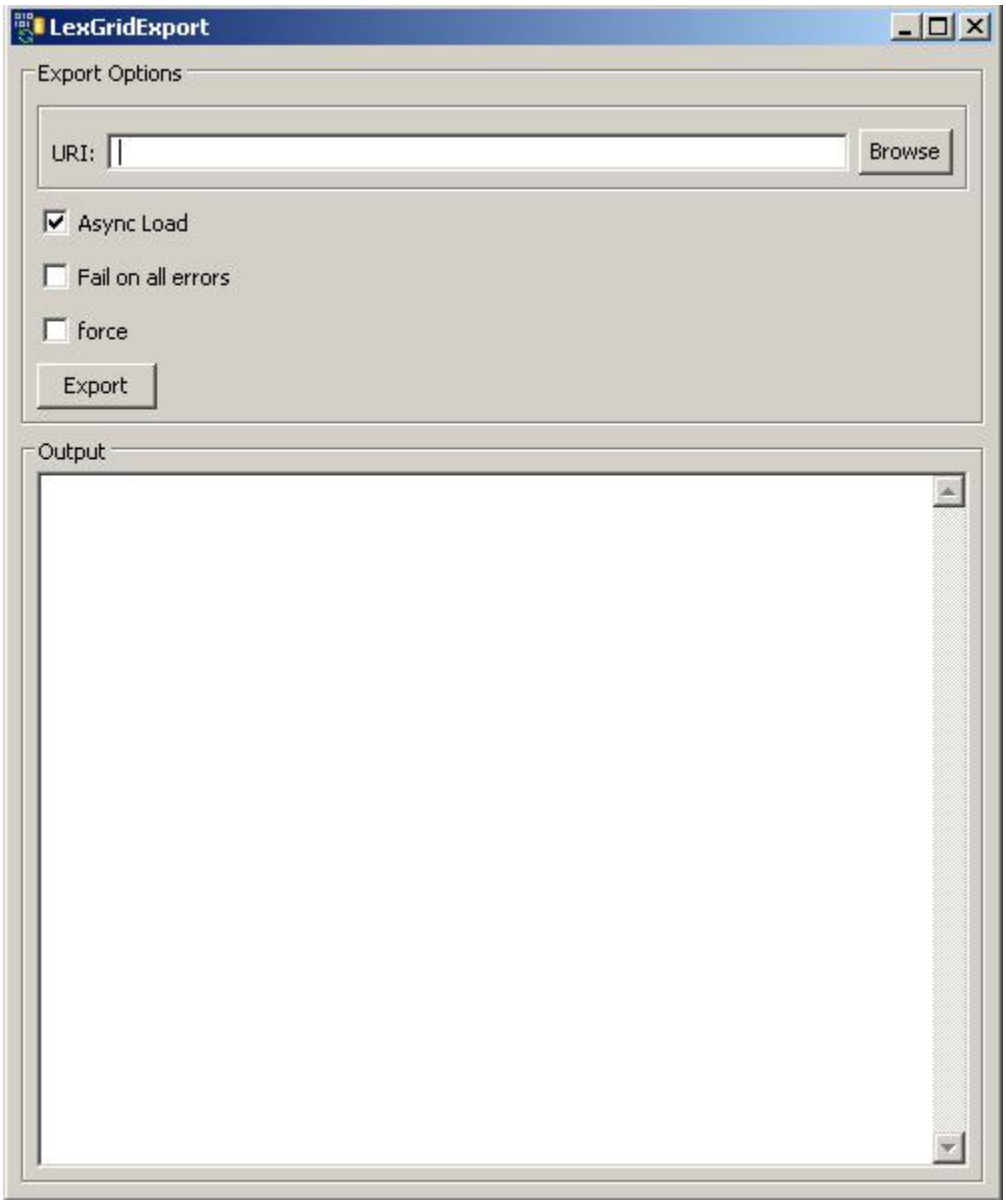


- *URI:* The path to MRMAP.RRF.
- *Manifest file*: Not implemented for the mapping loader.
- *Loader Preferences file*: Not implemented for the mapping loader.
- *Async load:* By default the loader loads as an asynchronous thread to other LexEVS activities allowing the JVM to continue if the loader fails.
- *Fail on all errors:* Load will fail on any source error if this is checked.
- *MRSAT file path:* The path to MRSAT.RRF.
- *Overwrite:* Overwrite current history for this source.
- *To Extend:* Not implemented for the mapping loader.
- *Load Post Processor (Extension name):* Not implemented for the mapping loader.

Export Terminology Menu

Export to selected formats from the LexGrid database

Menu Item	Menu Action
Export as OBO	This menu option will launch an exporter that exports the selected terminology into an OBO 1.2 format.
Export as OWL/RDF	This menu option will launch an exporter that exports the selected terminology into an OWL/RDF format.
Export as LexGrid XML	This menu option will launch an exporter that exports the selected terminology into the LexGrid XML format.



- *URI:* The path to output file.
- *Async load:* By default the loader loads as an asynchronous thread to other LexEVS activities allowing the JVM to continue if the loader fails.
- *Fail on all errors:* Load will fail on any source error if this is checked.
- *force:* Force an overwrite to an existing file if present. Otherwise an indication to stop if the destination file already exists.

Administrative Button Commands

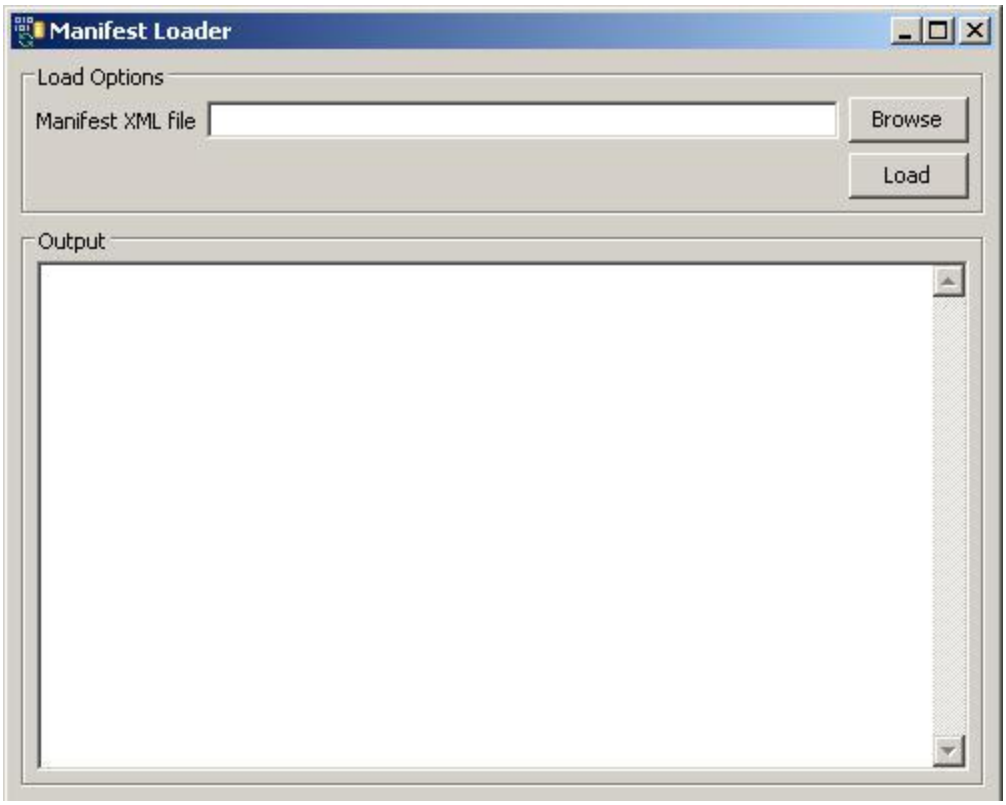
These are located in the lower right area of the top half of the LexEVS GUI.

Button	Button Action
Load Manifest	Brings up a dialog box to set up a manifest load.
Change Tag	Brings up a dialog that allows you to set (or remove) the tag on the selected terminology.
Activate	Activates the selected terminology. Only available if the terminology is currently deactivated.
Deactivate	Deactivates the selected terminology. Only available if the terminology is currently activated.
Remove	Deletes the selected terminology.
Remove History	Removes the NCI History data for the selected terminology. Only applicable to NCI Thesaurus terminologies.
Remove Metadata	Remove metadata load for a given coding scheme

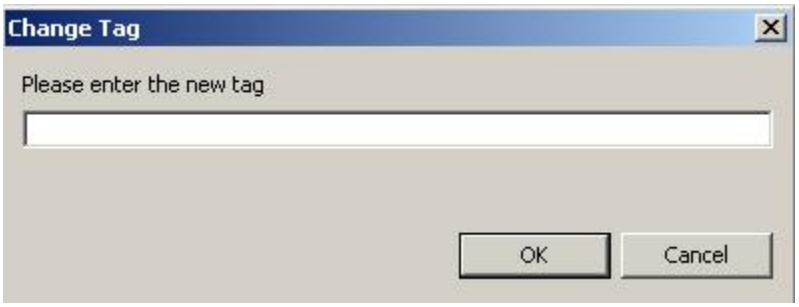
Rebuild Index	Rebuilds the internal indexes for the selected terminology. If no terminology is selected, rebuilds the indexes for all terminologies.
Run Postprocessor	Run a post processor algorithm against the selected coding scheme

Load Manifest

Manifest files are XML files that resolve against a LexGrid Model compliant manifest schema. They contain coding scheme metadata changes only. Once the target coding scheme is selected (and highlighted as a result) all that is required is that the user provide a path to a valid manifest file.



Change Tag



Simple Button Interface

User highlights the desired coding scheme and initiates the action by pressing a button:

Activate/Deactivate

Activate or deactivate a coding scheme. Coding Schemes which are not active cannot be queried or have some administrative actions performed upon them.

Remove

Deletes a given coding scheme and its indexes from the terminology service. If operating in single table mode removing a large terminology can be very time consuming.

Remove History

Remove a coding scheme history instance from the terminology service.

Remove Metadata

Removes a coding scheme metadata supplement from the terminology service.

Rebuild Index

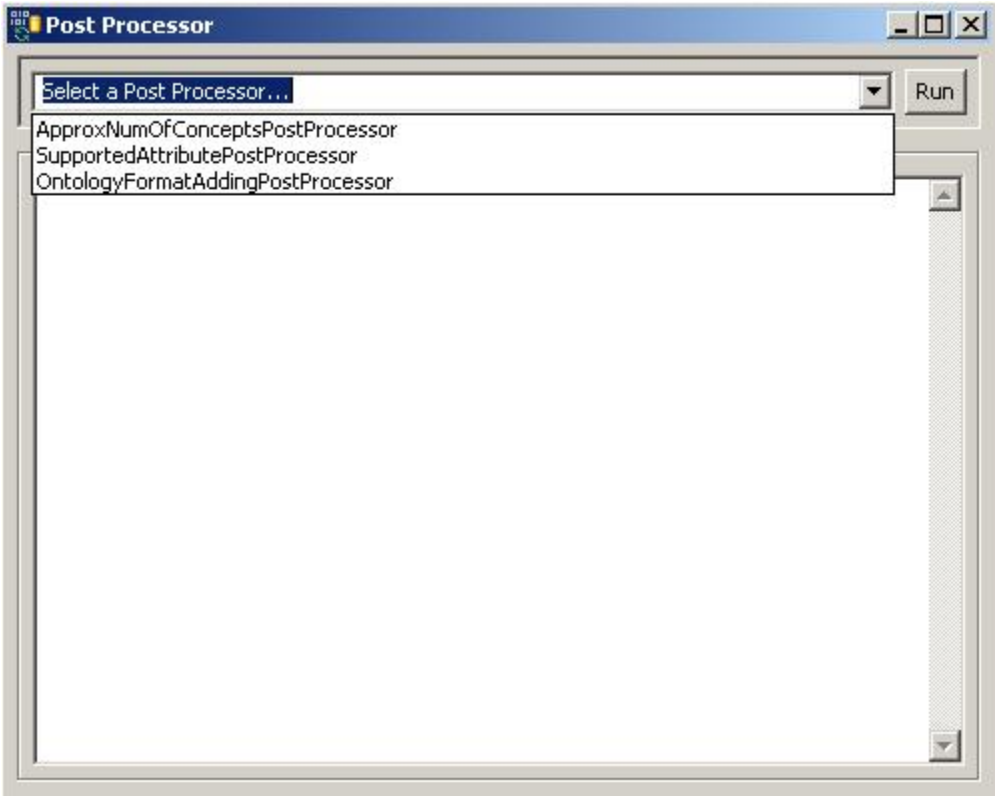
If a coding scheme is selected, it builds indexes for the given scheme. If not all coding schemes are reindexed. Reindexing a large set of sizable coding scheme could take some time.

Run Postprocessor

Executes a Loader post process. Loader post processes can be used to modify database content, do extra cleanup, or send notifications, for example.

> **Note**
>
> Post Process error/exception conditions will not effect Loader status.

Implementors can assume that database content has been loaded at the point of this call, but the load is not yet in a completed state and Lucene indexing has not been done.



1. Select the desired coding scheme
2. Click on the Run Postprocessor button
3. Select the post processor of choice
4. Run button runs the post processor

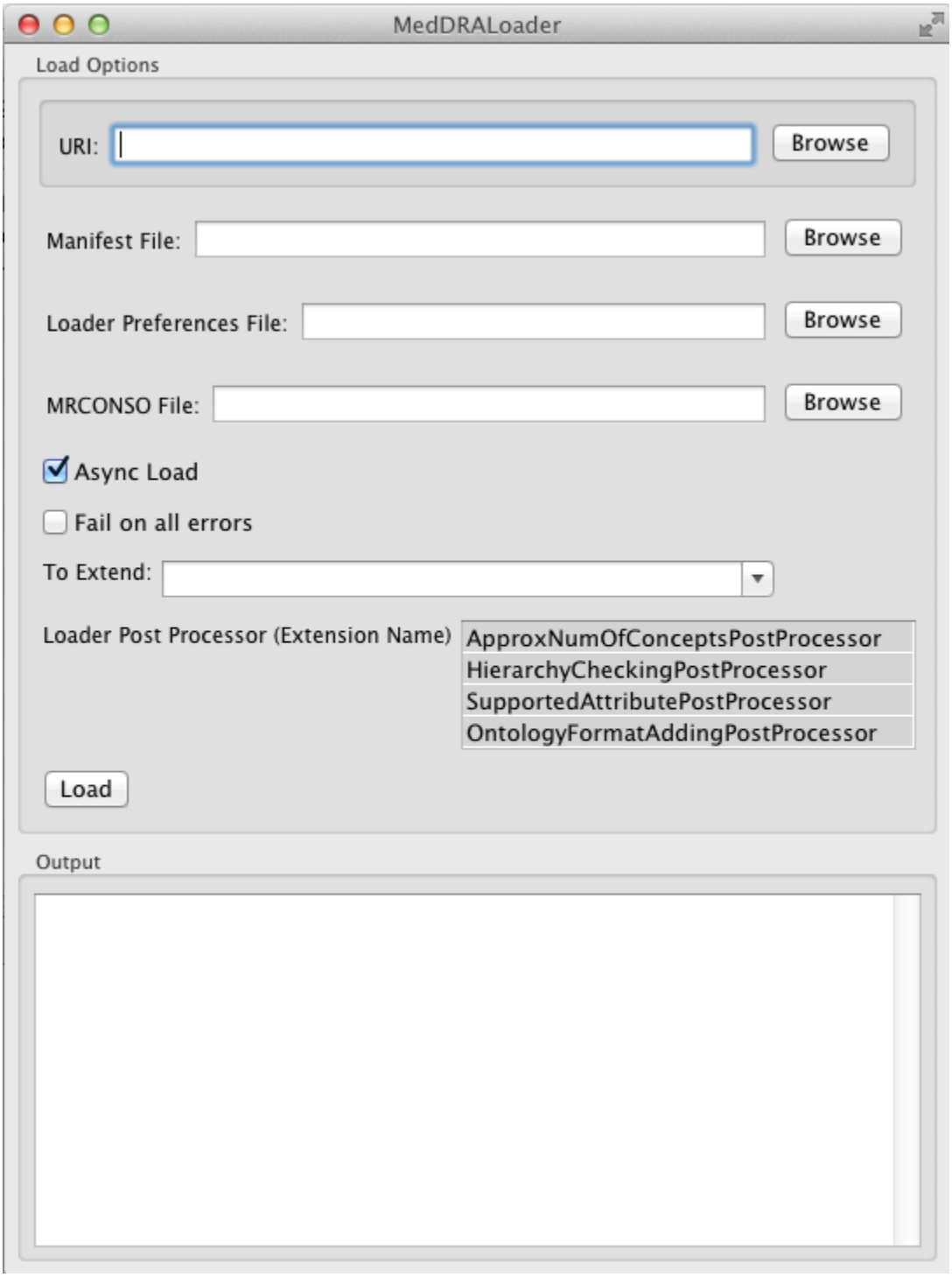
LexEVS 6.1 Administration Using the GUI Tool

LexEVS 6.1 features a number of new loaders. Most follow the typical pattern of use for LexEVS loaders. Alternative interfaces on the GUI tool include the following:

MedDRA Loader

Load MedDRA from proprietory MedDRA source:

Menu Item	Menu Action
Export as OBO	This menu option will launch an exporter that exports the selected terminology into an OBO 1.2 format.
Export as OWL/RDF	This menu option will launch an exporter that exports the selected terminology into an OWL/RDF format.
Export as LexGrid XML	This menu option will launch an exporter that exports the selected terminology into the LexGrid XML forma



- *URI*: The path to the file or folder containing source material for the load.
- *Manifest file*: The path to a manifest file that will change or add values to the coding schemes metadata
- *Loader Preferences file*: The path to a preferences file that will allow the user to decide how some source formats, such as OWL, will be loaded.
- *Async load*: By default the loader loads as an asynchronous thread to other LexEVS activities allowing the JVM to continue if the loader fails.
- *MRCONSO File*: Adding a URI for the MRCONSO.RRF source from the NCI Metathesaurus will load as properties the UMLS CUI's for MedDRA concepts.
- *Fail on all errors*: Load will fail on any source error if this is checked.

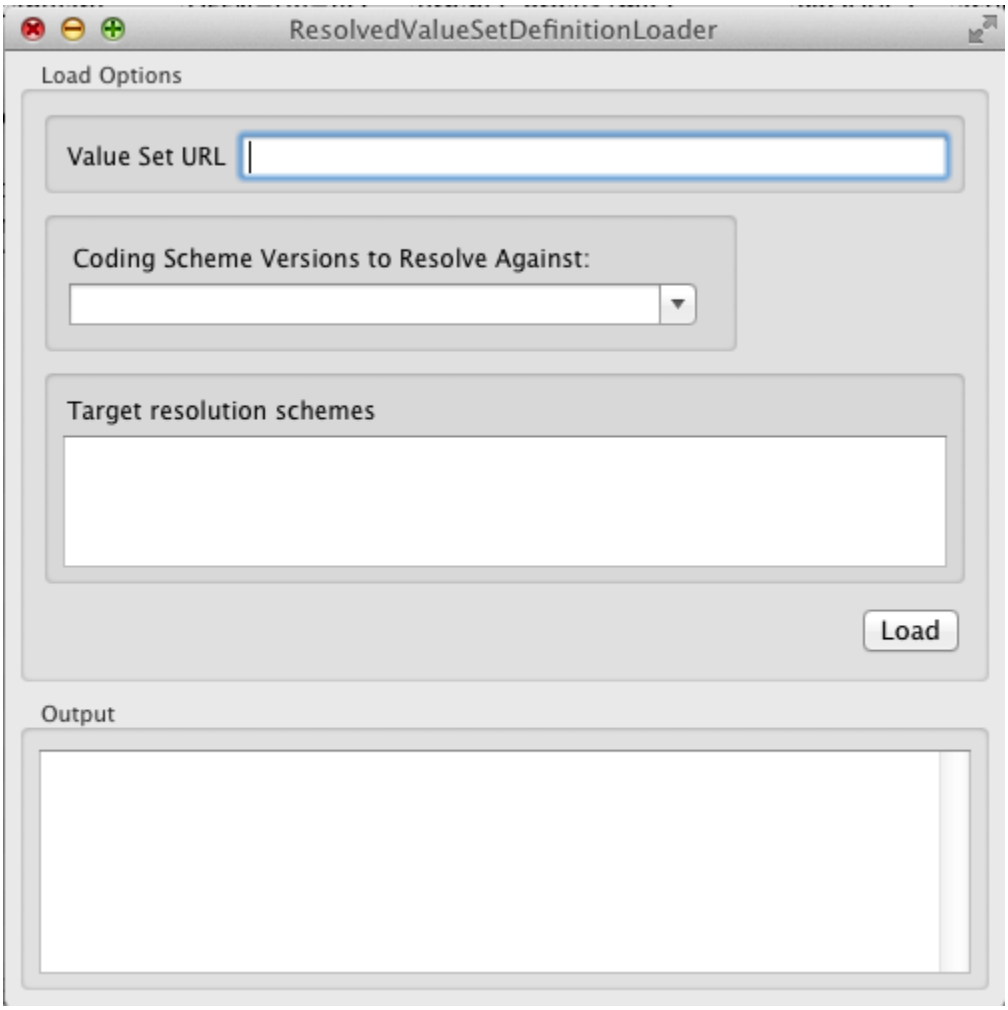
- *To Extend*: You may be loading this coding scheme as an extension to another. The drop down menu allows the user to make a choice between loaded coding schemes.
- *Load Post Processor (Extension name)*: The user may choose selected post processor algorithms to run on the coding scheme immediately after it loads.

MRCONSO Processing

MRCONSO.RRF is a large text file and a loading UMLS CUI's from it can degrade load time performance. We recommend pre-processing the file to restrict it to MedDRA content only. This can be accomplished easily in a UNIX/Linux environment with a command like the following: grep 'MDR' MRCONSO.RRF > MRCONSO.MDR

ResolvedValueSet Loader

Resolved value sets are loaded as coding schemes in LexEVS 6.1.



- *Value Set URL*: the URL or unique name of the Value Set Definition loaded in this LexEVS terminology service. (Must be loaded in the current service)
- *Coding Scheme Versions to Resolve Against*: Select the coding schemes in this terminology service that this value set definition will resolve against.
- *Target resolution schemes*: A list of schemes currently selected to resolve against. (Automatically populated by user selections from the previous drop down list.)