

# LexEVS FAQ

## Contents of this Page

- [End User FAQ](#)
  - [Where do I find the LexEVS log files?](#)
  - [Does LexEVS clean up the database after a failed load?](#)
  - [How does one set up a 5.X client that accesses your LexGRID services instead of having to load them locally into your own database?](#)
  - [Why do I get the following error during loading - ORA-00972: identifier is too long?](#)
  - [Loading UMLS SNOMED, how much time?](#)
  - [What is the difference between the Association Object attributes 'entityCode' and 'associationName'?](#)
  - [Is it necessary to load a mapping ontology as an ontology extension of the ontology being mapped?](#)
- [Developer FAQ](#)
  - [In short which is the right dir to load into Eclipse?](#)
  - [How do you debug LexBIG/LexEVS?](#)
  - [In addition to "contains", what other algorithms are available out-of-box for LexEVS grid service?](#)
  - [I was wondering if there was a way to use the LexEVS REST API to perform queries that returned concept codes based on search terms.](#)
  - [What is the best way to find out the current codingSchemeVersion for both the NCI\\_Thesaurus and NCI\\_MetaThesaurus?](#)
  - [Are there any changes in the resultSet schema depending on whether the codingSchemeVersion in the query has been set?](#)
  - [When using LexEVS, how do I get the number of resolved concepts without fetching all of them?](#)
  - [Is LexBIGService instance thread safe?](#)

## End User FAQ

### Where do I find the LexEVS log files?

For the LexEVS runtime you will find the log files here by default:

`<LexEVS Install Root>/logs`

unless this location was changed using the `<tt>LOG_FILE_LOCATION</tt>` entry in the config file called `<LexEVS Install Root>/resources/config/lbconfig.props`

where `<LexEVS Install Root>` by default is:

Linux: `/usr/LexGrid/LexEVS/<version>`

Windows: `\Program Files\LexGrid\LexEVS_<version>_`

and `<version>` is the version of LexEVS installed.

### Does LexEVS clean up the database after a failed load?

Yes - if LexEVS hits a non-recoverable error it will fail the load and clean up/drop any tables it was loading to.

There are exceptions to this. For instance, if the power goes out during the middle of a load, or the process ends suddenly for any other reason, the cleanup routine won't have a chance to be called and you will probably have to clean out the tables manually.

Thread: [Oracle error while loading UMLS](#)

### How does one set up a 5.X client that accesses your LexGRID services instead of having to load them locally into your own database?

Here's a few things to get you started:

The Distributed LexEVS/caGrid GUI (includes source in the zip):  
[5.0 Remote GUI zip](#)

Some general LexEVS 5.0 example code:  
[5.0 Example Code zip](#)

Small sample Distributed LexEVS Sample Client with source:  
[5.0 Distributed Sample Client zip](#)

Launch with:

```
ant runTestClient
```

Thread: [Questions on running a sample LexEVS program](#)

## Why do I get the following error during loading - ORA-00972: identifier is too long?

In your /resources/config/lbconfig.props file of your LexBIG installation, the 'DB\_PREFIX' value needs to be blank for Oracle --> DB\_PREFIX=

This is because Oracle puts a limit on table name length, and adding a prefix to the table names will exceed Oracle's limit.

Thread: [Loading UMLS causing Oracle error](#)

## Loading UMLS SNOMED, how much time?

The best thing to do to shorten up load time for a single ontology in UMLS is to subset the RRF files with MetamorphoSys. Dealing with a subset of the UMLS is much easier for the loader to deal with than the UMLS as whole.

You could try loading through one of the admin scripts. It does the same thing but you won't have to use the GUI.

Time will vary a lot depending on hardware and database config, so it's hard to estimate. You most likely won't gain much by adding ram – the bottleneck is usually how many inserts/second your database can handle. As a guideline, we load SNOMED (subsetting) in a dual quad core 32GB RAM in about 8 hours. This is with the db specially set up for fast inserts and tuned to what we're trying to do. The most important thing is to keep an eye on the logs and make sure progress is still being made, as it is hard to estimate how much time it will take.

Thread: [Loading UMLS SNOMED, how much time](#)

## What is the difference between the Association Object attributes 'entityCode' and 'associationName'?

For 5.x, an Association is a type of Entity – meaning it has an EntityCode, EntityCodeNamespace, etc. It can also contain Properties and participate in relationships with other Entities. An 'Association' is a special type of Entity, whereas its job is to describe relationships that Entities may have to each other. It can describe the forward and reversion directional names, flags to indicate whether the Association is Transitive or not, and so on.

The 'associationName' is simply the name given to the relationship.

So:

Father -> 'hasChild' -> HisSon

In this case 'hasChild' is the 'associationName'

But, in LexEVS 5.x an Association is also an Entity, so it must also have an 'entityCode'. The 'entityCode' can (and often is) the same as the 'associationEntity', but it does not have to be. It is simply the unique identifier of the Entity – the 'entityCode' can be any String.

## Is it necessary to load a mapping ontology as an ontology extension of the ontology being mapped?

A "mapping" ontology should never be an "extension" of one of the ontologies it is mapping. By default, a "mapping" ontology imports the contents of the mapped ontologies automatically.

Thread: [Loading a map as and extension to a vocab changes restrictToCodes\(\) behavior](#)

## Developer FAQ

### In short which is the right dir to load into Eclipse?

If you're looking to bring everything into Eclipse, I would recommend checking out from SVN – because that way you'll get all the Eclipse metadata type files with it. You should be able to use anonymous checkout for these. Here's the projects you'll need to check out, each as an individual Eclipse project:

- Indexer
- lbAdmin
- lbExamples
- lbGUI
- lbImpl
- lbInterfaces
- lbModel
- lbPackager
- lgConverter
- lgModel
- lgModel.emf
- lgResourceReader
- lgSharedLibraries
- lbUtil

Thread: [LexEVS/LexBIG source tree question](#)

## How do you debug LexBIG/LexEVS?

As for debugging, there's a couple of different ways to go about that. First, you can run the GUI from Eclipse. When you checked things out you should have gotten some run configurations named "LB\_GUI" or something similar. There's various ones for different OS's. What's handy about that is the GUI does a pretty good job of reflecting our API, so you can work through API calls and set breakpoints as you go. When I want to debug something, sometimes I just start the GUI in debug mode and then put breakpoints as needed.

Another thing I sometimes do is just create another eclipse project (say, 'LexEVSTestProject') in my Eclipse environment. The key here is to include the 'lblemp' as a required project. This will pull in all the LexEVS classes and config files into your classpath. Then, you can write code in your test project and debug that way.

Thread: [LexEVS/LexBIG source tree question](#)

## In addition to "contains", what other algorithms are available out-of-box for LexEVS grid service?

There are a list of algorithms with a short description in the "Programmer's Guide", Section "Search Algorithms"

Thread: [List of LexEVS Algorithms Available](#)

## I was wondering if there was a way to use the LexEVS REST API to perform queries that returned concept codes based on search terms.

General LexEVS REST Documentation can be found in the "Programmers Guide," Section "XML-HTTP API"

Thread: [Question on Using LexEVS REST API](#)

Thread: [XML-HTTP API for LexEVS](#)

Thread: [LexBIG 5.0 REST API](#)

## What is the best way to find out the current codingSchemeVersion for both the NCI\_Thesaurus and NCI\_MetaThesaurus?

You could use this to get information about the CodingScheme itself. The representsVersion will display the version for the vocabulary

```
http://lexevsapi.nci.nih.gov/lexevsapi50/GetXML?query=org.LexGrid.codingSchemes.CodingScheme&org.LexGrid.codingSchemes.CodingScheme[@_codingSchemeName=NCI_Thesaurus]
```

Thread: [HTTP REST Interface: Attribute startIndex Vs. statIndex?](#)

## Are there any changes in the resultSet schema depending on whether the codingSchemeVersion in the query has been set?

If the codingSchemeName AND version are both set, the system does not need to retrieve all of the results at once – some collections can be lazy loaded (for instance, the Presentation List, Definition List, etc). In this case, a link will be provided to access the results, and the results themselves will not be loaded immediately. If just the codingSchemeName is provided, the system can't lazy load anything (because just a codingSchemeName is ambiguous – there may be several different versions of a loaded codingScheme), and all results are returned at once. This is done for performance reasons, specifically because loading all Properties of a large amount of Concepts can be expensive.

Thread: [resultSet on setting/unsetting codingSchemeVersion](#)

## When using LexEVS, how do I get the number of resolved concepts without fetching all of them?

You could resolve the CodedNodeSet to a ResolvedConceptReferencesIterator – then call 'numberRemaining()' on the iterator to get the number. This way, you won't have to resolve everything all at once.

Thread: [Getting only number of resolved concepts?](#)

## Is LexBIGService instance thread safe?

For example:

I have a servlet and I'd like to initialize the only instance of LexBIGService in init() method. And then, in subsequent HTTP calls, use this instance. I do initialization in following way:

```
appSvc = (LexBIGService) ApplicationServiceProvider.getApplicationServiceFromUrl(url, "EvsServiceInfo");
```

Yes, that should be fine – there are a few things to note however.

Usually in a caCORE SDK system when you call 'ApplicationServiceProvider.getApplicationServiceFromUrl(url, "..."); more than once with the same parameters, you'll just get the same object returned to you (at least in caCORE SDK 4.0). This happens because its basically set up as a Spring singleton-scoped bean. We had to stray from that a little because we allow the user to store a list of registered security tokens for secured vocabs on the client side. Because of this, we can't have the LexBIGService be a singleton because then if a different user called ApplicationServiceProvider.getApplicationServiceFromUrl(url, "..."); from the same VM, they'd get all the other users' registered security tokens, which we don't want. So how it works for us is every call to 'ApplicationServiceProvider.getApplicationServiceFromUrl(url, "..."); returns a Prototype-scope (i.e. new instance) Proxy of LexBIGService.

So, yes, calling appSvc = (LexBIGService) ApplicationServiceProvider.getApplicationServiceFromUrl(url, "EvsServiceInfo"); in some sort of init() method or factory would be fine – as long as everything that is using that LexBIGService is ok with sharing the same security tokens.

A scenario where this wouldn't work out is if you had a web site where a user could log in, register some security tokens... etc. You would want to create a new LexBIGService for every user so that you would be sure that their security tokens would stay independent of each other. In other words, make the LexBIGService "session" scoped.

Thread: [LexBIGService instance thread safe?](#)