National Cancer Institute

U.S. National Institutes of Health | www.cancer.gov

### caBIG' Knowledge Center
A part of the Enterprise Support Network

👤 Log in    create account    account help

| Home | Knowledge Centers | Discussion Forums | Bugs/Feature Requests | Development Code Repository |
|------|------|------|------|------|

**page**    discussion    view source    history

# Installing LexEVS in a web-based environment

Installing LexEVS with its LexGrid Model/DB (back end) > Installing LexEVS in a web-based environment

## Introduction

This document is a section of the Installation Guide.

This section provides requirements and instructions for installing LexEVS in a web-based environment.

## Minimal system requirements

- Internet connection
- 2 GB RAM
- Tested Platforms (Similar Hardware Specification for Operating System)

LexEVS has been tested on the platforms shown in Table 1.

|  | *Linux Server* | *Linux Server* | *Windows* |
|------|------|------|------|
| *Model* | HP Proliant DL 380 | Penguin | Dell Latitude |
| *CPU* | 2 x Intel® Xeon™ Processor 2.80GHz | Dual AMD Opteron 248 processors (64 bit) | 1 x Intel® Pentium™ Processor 2.00GHz |
| *Memory* | 4 GB | 16Gb | 1.5Gb |
| *Local Disk* | System 2 x 36GB (RAID 1) Data = 2 x 146 (RAID 1) | 250 GB Raid 1 disk drive(s) 250 GB stand along disk drive | System 1 x 80GB |
| *OS* | Red Hat Linux ES 3 (RPM 2.4.21-20.0.1) | Fedora Core 3 (64 bit) OS | Windows XP Professional |

*Table 1 - Platform Testing Environment*

## Software requirements

### Required Software - Not Included in LexEVS

You must download and install the required software that is not included with LexEVS (listed in Table 2). The software name, version, description, and URL hyperlinks (for download) are indicated in the table.

(Required software that *is* included with the LexEVS is listed in Appendix I)

| Software Name | Version | Description | URL |
|------|------|------|------|
| Java Software Development Kit (SDK):Java 2 Standard Edition (J2SE) | j2sdk1.5.0_04 or higher | The J2SE Software Development Kit (SDK) supports creating J2SE applications | http://java.sun.com/javaee/downloads/ 🖻 |
| MySQL Database* | MySQL (5.0.45) or | MySQL 5.0 Community Edition | http://downloads.mysql.com/archives.php? |

---

**vocabkc contents**
- Main Page
- What's New
- Forums
- Bugzilla
- Code Repository
- Feedback
- Contact Us

**tools**
- LexBIG/LexEVS
- LexWiki
- NCI Protégé
- Related Tools and Models

**projects**
- LexAjax
- LexGrid
- Cancer Data Standards Repository (caDSR)
- Common Terminology Criteria for Adverse Events (CTCAE)
- Open Health Natural Language Processing (OHNLP) Consortium
- Ontology Development and Information Extraction (ODIE)

**semantic infrastructure**
- SI Main Page
- Initiatives
- Requirements

**other resources**
- Library of Documents
- Documentation and Training for Tools
- Index of Terminologies
- Standards and Standards Influencing Organizations
- Outreach

**external links**
- VCDE Workspace
- caBIG® Community Website
- caBIG® Support Service Providers

**help**
- Editing Wiki Pages

| | | | p=mysql-5.0&v=5.0.45 🖉 |
|---|---|---|---|
| PostgreSQL* | 8.x or higher | Open source relational database management system | http://www.postgresql.org/ 🖉 |
| *MySQL or PostgreSQL installation is required. | | | |

*Table 2 - Required software and technology for the LexEVS*

> **NOTE:** Some JDBC drivers are included with LexEVS installer others will have to be downloaded by the user due to licensing considerations. These drivers are placed in the `{LEXEVS_DIRECTORY}/runtime-components/extLib` directory.

## Installation

Since LexEVS is server oriented software, a common deployment scenario is to package LexEVS into a WAR file and deploy it into JBoss or another Application Server. The following are recommendations for packaging and configuring LexEVS for JBoss usage.

If not deploying into an Application Server such as JBoss, this chapter can be skipped. While these instructions are tailored to JBoss, they should serve as a reasonable guide in packaging for other Application Servers.

Complete a standalone installation of LexEVS before starting this section. These files will be copied to create your JBoss package.

### Creating the WAR file

To add LexEVS functionality to a WAR file, add the `lexbig.jar` file to the lib folder of the WAR file. Adding the `lbRuntime.jar` file to the lib folder may result in the duplication of many common libraries that LexEVS and JBoss share. In particular errors will occur due to a collision between the log4j libraries that LexEVS and JBoss both use.

The `lexbig.jar` can be found in the `'runtime-components'` folder of the LexEVS installation.

The `lexbig.jar` file adds LexEVS runtime functionality to JBoss; however, 3rd party dependencies must be added to the WAR file lib folder as well. These jar files are located in the `'runtime-components/extLib'` subfolder shown below.
**Note that the screenshots are examples provided for clarification. The version numbers on your filenames may be different.**
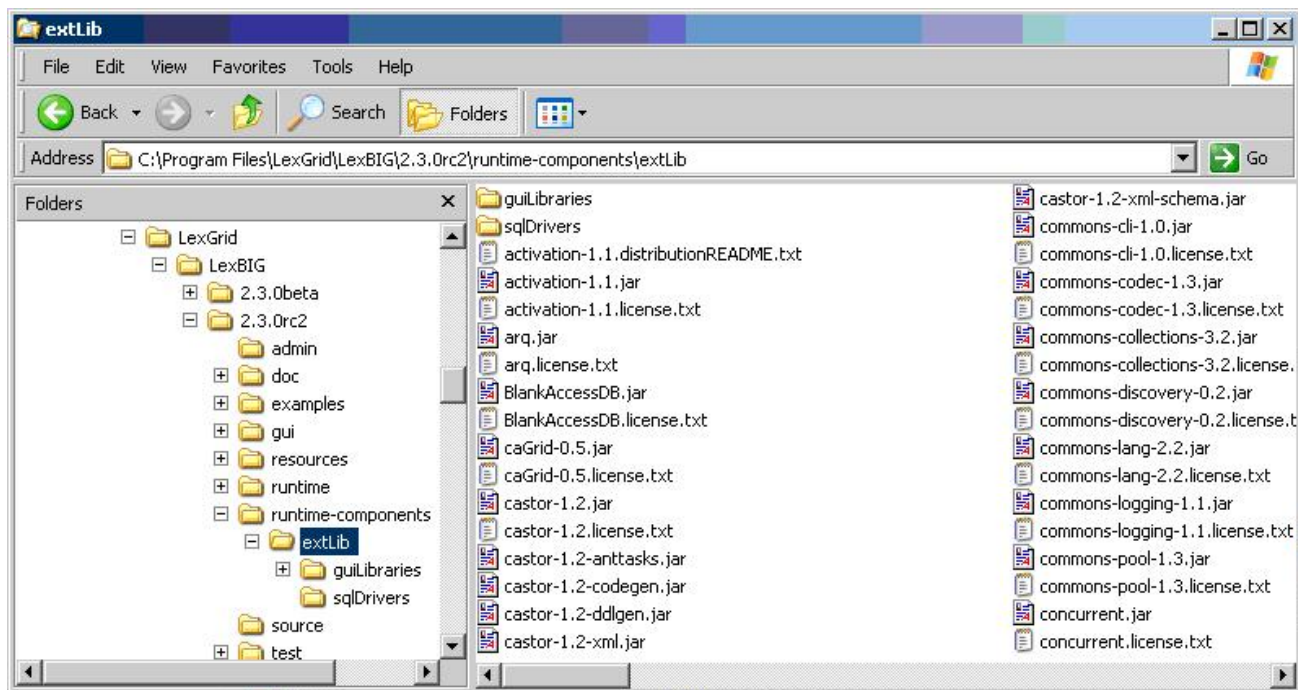


Avoid duplicating jars already contained in the WAR package or jar files that JBoss already provides. For example, JBoss already provides a log4j `x.x.x.jar` – this file should NOT BE COPIED into the WAR package. **Note:** this jar is singled out because it is known to cause errors in JBoss if you deploy a WAR file that contains a log4j library.

Do not copy files from `'runtime-components/extLib/guiLibraries'` or any of the sql drivers in the `'sqlDrivers'` folder. The gui libraries

are not needed and the sql drivers will be dealt with later.

## Creating the file storage location for the LexEVS runtime

LexEVS runtime needs to have the ability to read and write from a local file system in order to function properly. The files that it writes also need to persist through a stop and start of the JBoss server. These files cannot be placed inside of the WAR file, since JBoss extracts WAR files into a differently named temporary folder every time it starts up.

Create a single folder to store this information, and configure the LexEVS JBoss instance to use this folder while it is running for all of its file system access needs.

Below is a screenshot of the file structure to create:
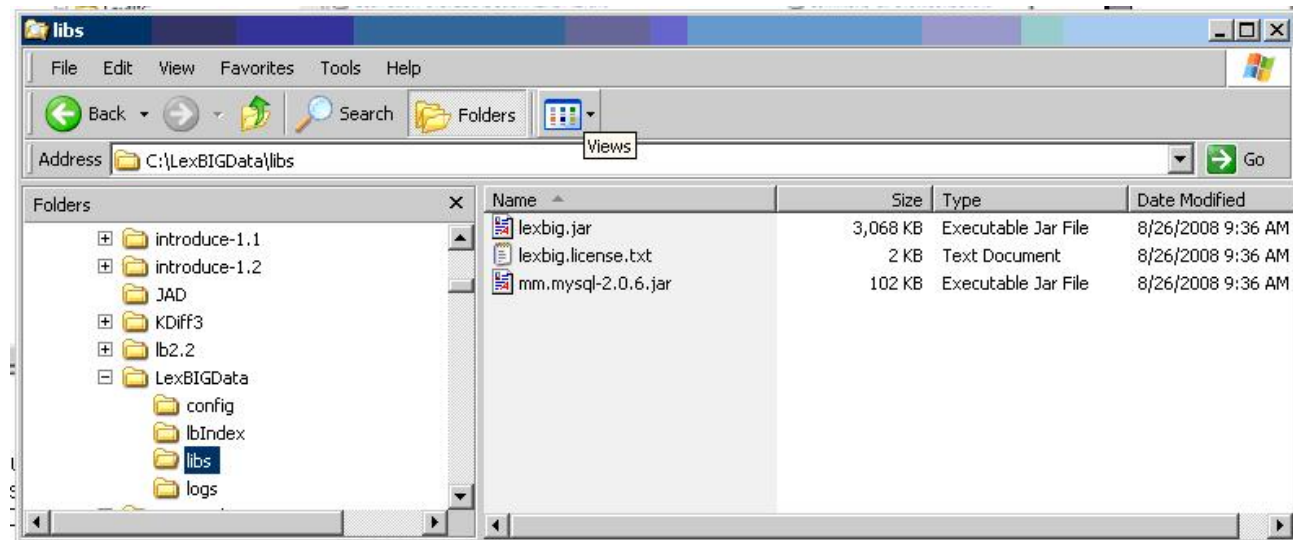


Create a top level folder named `LexBIGData`. Create sub-folders in this directory named `config`, `lbIndex`, `libs`, and `logs`.

The folder `config` is used for storing the LexEVS configuration, storing the terminology registry, and storing lock files. Copy `LexEVS <version>\resources\config\config.props` into this folder now. This file will be customized later.

The folder `lbIndex` is where LexEVS will store the indexes that it creates. Leave this folder empty.

In the folder `libs` copy the SQL drivers. You may have to download drivers compatable with the database verion your LexEVS application connects to. Any LexEVS extensions can also be placed here. Copy the jar `LexEVS 5.x\runtime-components\lexbig.jar` into this folder. A duplicate of this jar file needs to be here to assist with some custom classloader work that bypasses the normal classpaths.

The folder `logs` is where LexEVS will write its log messages. Leave this folder empty.

Your directory structure should now look like the screen shot above.

## Configuring lbconfig.props for JBoss deployment

There are a few variables in the lbconfig.props file in your LexBIGData folder that will need to be changed for the JBoss deployment. There are a number of ways that this file can be configured – but these instructions set up directories relative paths – allowing the WAR file to be easily moved from one server to another.

The lbconfig.props file should already be configured for standalone mode at this point, as described earlier in the manual. This section will only describe the changes that should be made for JBoss deployment.

There following five variables need to be set to the following values in the lbconfig.props file.

LG_BASE_PATH= This variable is being left blank. This will cause LexEVS to automatically set this variable to the location of the lbconfig.props file when it starts up. All other paths in the lbconfig.props file can then be set relative to the folder that contains the lbconfig.props file that was found when the system started up – and it should use the lbconfig.props file in your LexBIGData folder.

JAR_FILE_LOCATION=../libs/ This variable is being set to point to the `libs` folder that we created in the LexBIGData folder. The two periods tells it to go up one folder from the place where it started (the config folder) and then down into the libs folder.

REGISTRY_FILE=registry.xml By only putting a file name here, a file of that name will be created in the same folder where it started (the config folder in the LexBIGData folder).

INDEX_LOCATION=../lbIndex Sets the index storage location to the lbIndex folder in the LexBIGData folder.

LOG_FILE_LOCATION=../logs Sets the log location to the logs folder in the LexBIGData folder.

## Locating the file storage location for the LexEVS runtime

Now that you have created your `LexBIGData` folder, and configured the config.props file, you need to decide where to place it on the

system where JBoss and your WAR file will be deployed. Location is up to the user – so long as it is permanent. A couple of possibilities include keeping it with JBoss -- in the JBoss data foler,, `\…\jboss-4.0.4.GA\server\default\data\LexBIGData,'` or it could go into a typical Linux application folder such as `/apps/LexEVS/LexBIGData/`.

## Locating the lbconfig.props file when JBoss starts

The only step remaining now is to help LexEVS find the proper `lbconfig.props` file when it initializes. There are a number of ways that this could be done – but the easiest way is to set the Java System Variable `LG_CONFIG_FILE` to the location of the lbconfig.props file.

This can be done by modifying the script that launches JBoss – adding the following (modify as appropriate for the place that you picked for your `LexBIGData` folder)

`-DLG_CONFIG_FILE=C:\Program Files\jboss-4.0.4.GA\server\default\data\LexBIGData\config\lbconfig.props`

Alternatively, you could programmatically set this system variable from your application when it starts up and before it makes its first LexEVS call.

Read the documentation in the lbconfig.props file for other `LG_CONFIG_FILE` configuration possibilties.

**NOTE:** JBoss tends to run out of memory while running with the default settings of JBoss. To fix the problem, modify the run.conf file under JBOSS_home/bin to change the memory setting and how frequently the rmi garbage collector runs, by editing the JAVA_OPTS line as follows:

```
if [ "x$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="-server -XX:MaxPermSize=128m -Xms256m -Xmx2000m -Dsun.rmi.dgc.client.gcInterval=120000 -
Dsun.rmi.dgc.server.gcInterval=120000"
fi
```

## Moving a terminology to another SQL server

In some deployment scenarios, it may be necessary to move a very large terminology onto a different SQL server than the standard LexEVS server configured above for scalability reasons.
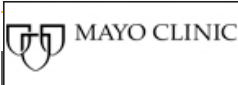
This is supported, but there are some limitations and cautions:

- Manually edit the `registry.xml` file (It may have an alternate name as denoted in the `REGISTRY_FILE` variable of `lbconfig.props`) with a text editor.
- This terminology cannot be removed using the LexEVS API. To remove the terminology, edit the `registry.xml` file again, and manually drop the database from the server. This leaves an orphaned index -- running the orphaned resources clean up tool wipes the index and associated entries.
- This "extra" SQL server must use the same driver and same username and password as the default LexEVS server since the configuration file only supports a single password and username set.

Steps to move a terminology to a new server:

1. Manually move the proper database (or proper subset of tables in `SINGLE_DB_MODE`) to the new server. The LexEVS `TransferScheme` tool (in the admin scripts folder) can help you determine the proper SQL databases, tables, and commands that you will need to use to accomplish this. Alternatively, you can look at the `registry.xml` file to determine what database and/or tables you need to move.
2. Manually edit the registry.xml file. You will need to change the value of the dbURL parameter on the proper line for this terminology that you are using.
3. Restart LexEVS, and ensure that the terminology is still available.

Categories: VKC Contents | Documentation | LexEVS