

3 - Installing LexEVS 5.x Local Runtime

Contents of this Page

- [Introduction](#)
- [Preliminary Considerations](#)
- [Downloading and Installing the LexEVS Local Runtime](#)
- [What's Inside](#)
- [Configuring the LexEVS Environment](#)
 - [MySQL Properties](#)
 - [PostgreSQL Properties - Modifying the postgresql.conf File for PostgreSQL](#)
 - [Server Properties](#)
- [Testing the LexEVS Local Runtime](#)
- [Troubleshooting](#)

Introduction

This document is a subsection of the [Installation Guide](#).

This section provides requirements, instructions, and troubleshooting information for installing a LexEVS Local Runtime.

Preliminary Considerations

Before You Begin Warning

LexEVS has been tested with the operating systems and hardware specified in the LexEVS 5.x Supported Platforms]. While LexEVS is expected to run on many variations of hardware and software similar to the test platforms, results cannot be guaranteed. Before you begin you should check to make sure that your platform will support the software.

- There are a number of [LexEVS 5.x Prerequisite Software Supported](#) products that you must install to run a LexEVS Local Runtime on your platform. You must make sure that you have the proper software and versions, depending on which LexEVS environment you are installing. Do this now, for the Local Runtime designated prerequisite software, before continuing to the install steps.
- As an option you may consider the command line install of a LexEVS Local Runtime for putting all the necessary files in place based on a pre-decided list of components to include. As noted in the last step of the GUI Installer for LexEVS Local Runtime, you can save a file that represents the choices you have made during the GUI install. This file can be used as-is or be edited and then fed into an install on another server. Follow the steps for [6 - Installing LexEVS 5.x and Higher Local Runtime \(Optional Command Line Method\)](#) if you should choose this route.
- Complete the [#Downloading and Installing the LexEVS Local Runtime](#) steps below.

Once you have completed the prerequisite software install, the Local Runtime environment install steps, and the verification test as described in this guide then you should be ready to start programming using the API to meet the needs of your application. Not counting prerequisite software products the installation and verification should not take more than 60 minutes.

Downloading and Installing the LexEVS Local Runtime

Step	Action
------	--------

1. Download the latest version of the Local Runtime GUI install package from the right. The location you have chosen to save this on your computer will be referred to as the `SAVE_DIRECTORY`. Our command examples will use scratch as this directory. If you plan to install the LexEVS Remote API or the LexEVS Grid Services in conjunction then the Local Runtime version must match what you plan to install.

Note

Access to the downloads does not require an account. If you are having trouble downloading files then you may need to disable pop-up blockers or check any firewall settings at your site.

GUI Installer for LexEVS Local Runtime

A package for use as a single download with everything you need to install the Local Runtime.

- 5.1.0 [LexEVS-install-5.1.jar](#). Download the [Readme](#) for the latest information for about this release.
- 5.1.4 [LexEVS-install-5.1.4.jar](#). Download The [Readme](#) for a list of the fixes and functions and installation instructions for upgrading an existing install.

LexEVS Local Runtime Release Notes

An optional download if you want to see the resolved issues and enhancements provided in the primary release. Readmes will contain information for fix releases.

- [lexevsapi51_notes.html \(wiki\)](#)

LexEVS Local Runtime Command Line Install

An optional download for installing the Local Runtime without using the GUI installer. This file can be generated by the GUI installer or downloaded here.

- 5.1.0 [LexEVS-install-config-5.1.xml](#)

LexEVS Local Runtime Environment

Includes the core runtime, LexEVS API, loaders, and administrative utilities. This file is included with the GUI installer above.

- [LexEVS_51_localRuntime.jar](#)

LexEVS Local Runtime 3rd Party Dependencies

Includes code from other open source projects required by the LexEVS API. This file is included with the GUI installer above.

- [LexEVS_51_localRuntime_dependencies.jar](#)

2. Using a file explorer, navigate to the `SAVE_DIRECTORY`. Double Click on the downloaded JAR file. This will launch the install wizard.

As an alternative to a file explorer, use a command prompt. Change to the `SAVE_DIRECTORY`:

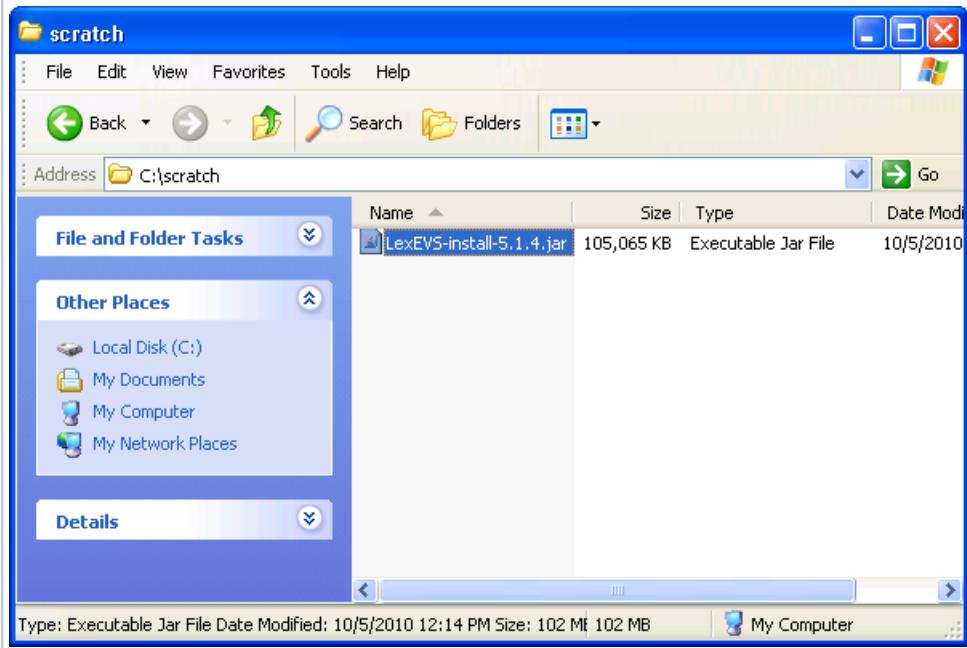
```
cd {SAVE_DIRECTORY}
```

Then enter the command to begin the installation wizard:

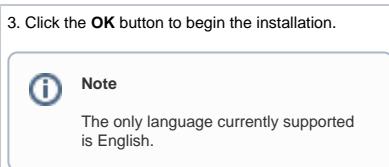
```
java -jar LexEVS-install-{version}.jar
```

...where `version` is the version you downloaded
For example:

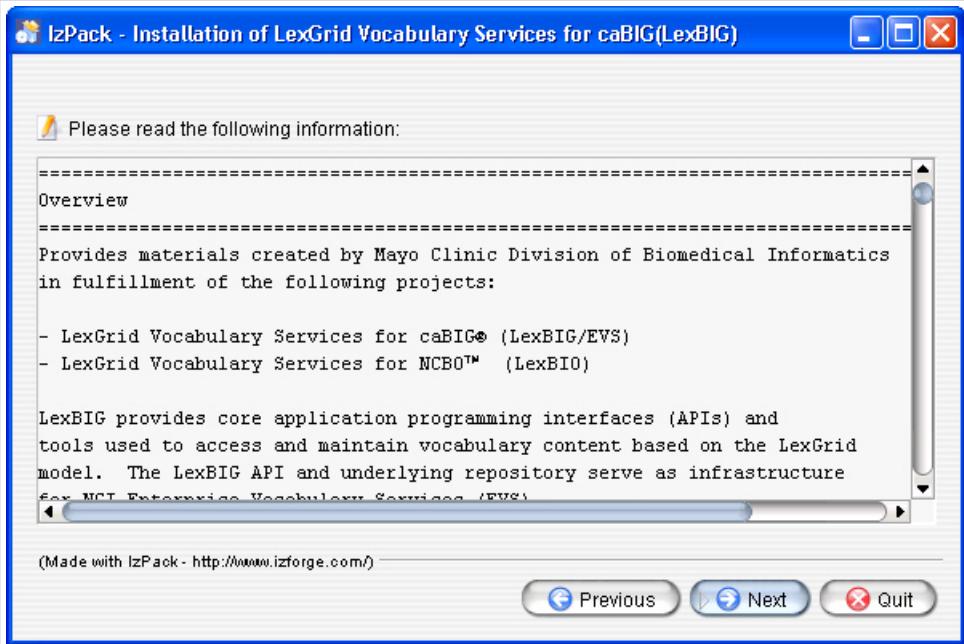
```
java -jar LexEVS-install-5.1.jar
```



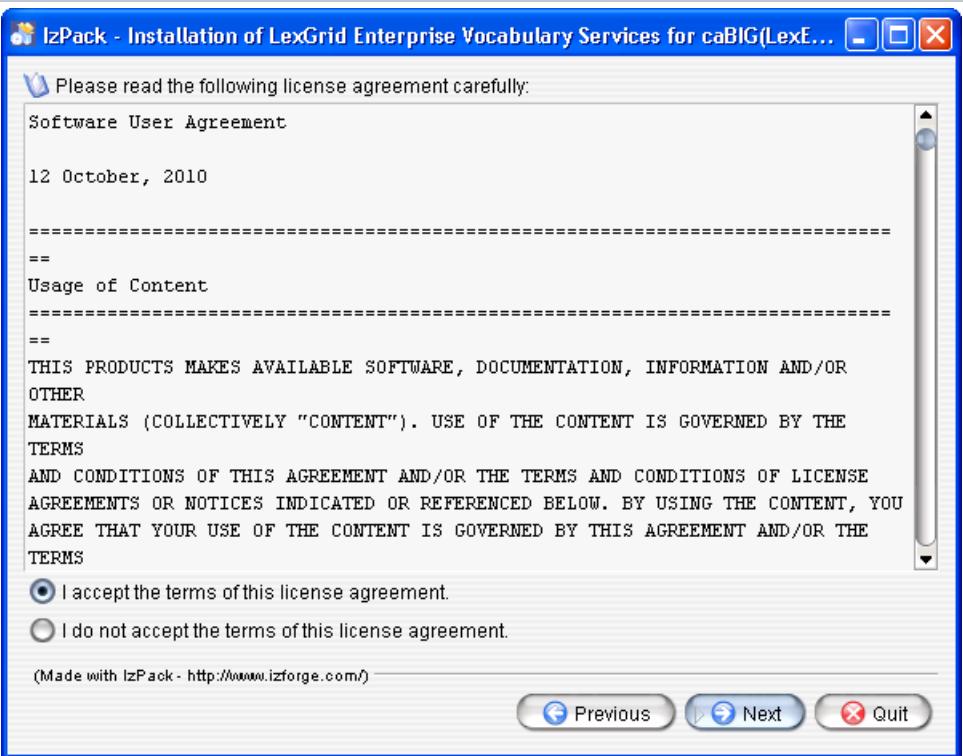
3. Click the **OK** button to begin the installation.



4. After the initial welcome screen, the release notes for the LexEVS distribution are displayed. Once you have read through the release notes click the **Next** button.



5. Review the license agreement of the LexEVS software. Select "I accept the terms of this license agreement." and click **Next**.



6. Enter the path where you would like the LexEVS software installed. This will be referred to as **LEXEVS_HOME** throughout the remaining instructions.

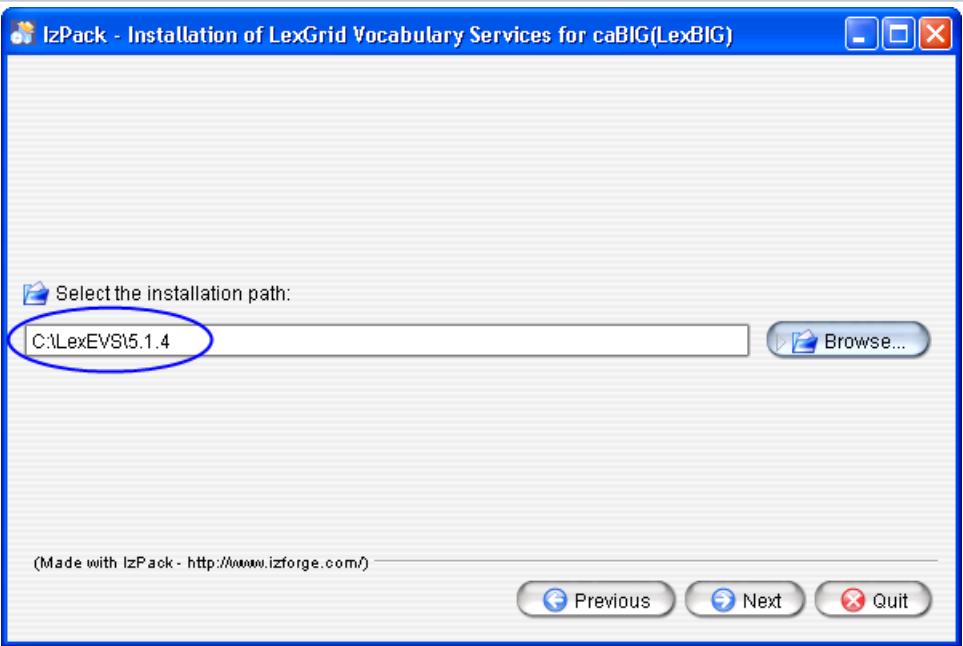
Note

In any operating system it is **NOT** recommended to use spaces in the install path. You will be better off if you do not use the default "Program Files" path in Windows.

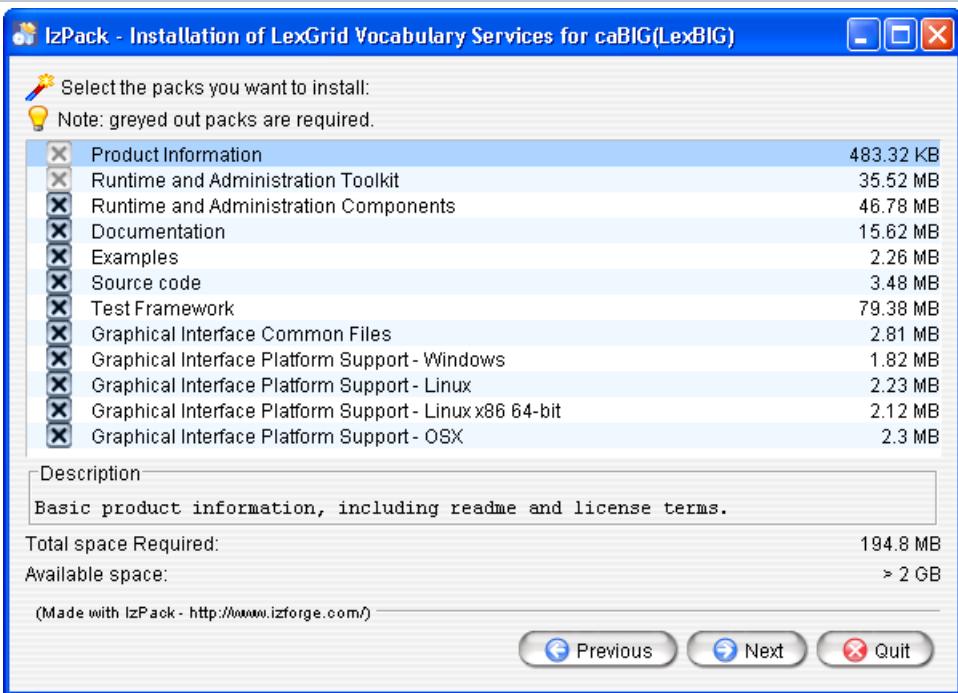
Click the **Next** button.

Note

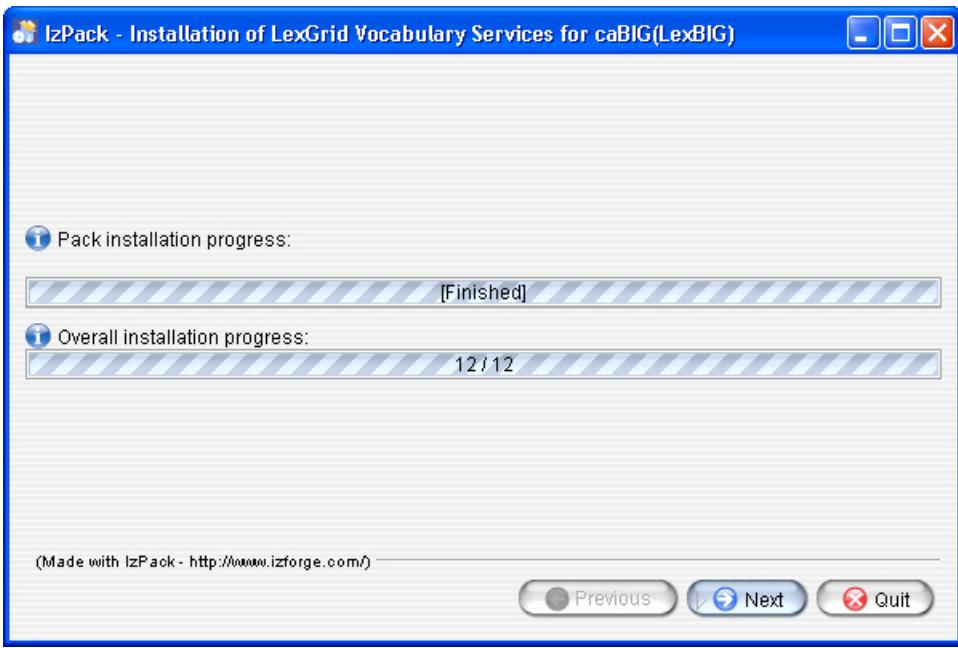
If the directory does not exist, the program will prompt to proceed with creating the new directory. If the directory does exist, the program will prompt to overwrite the directory and files that may be contained within it.



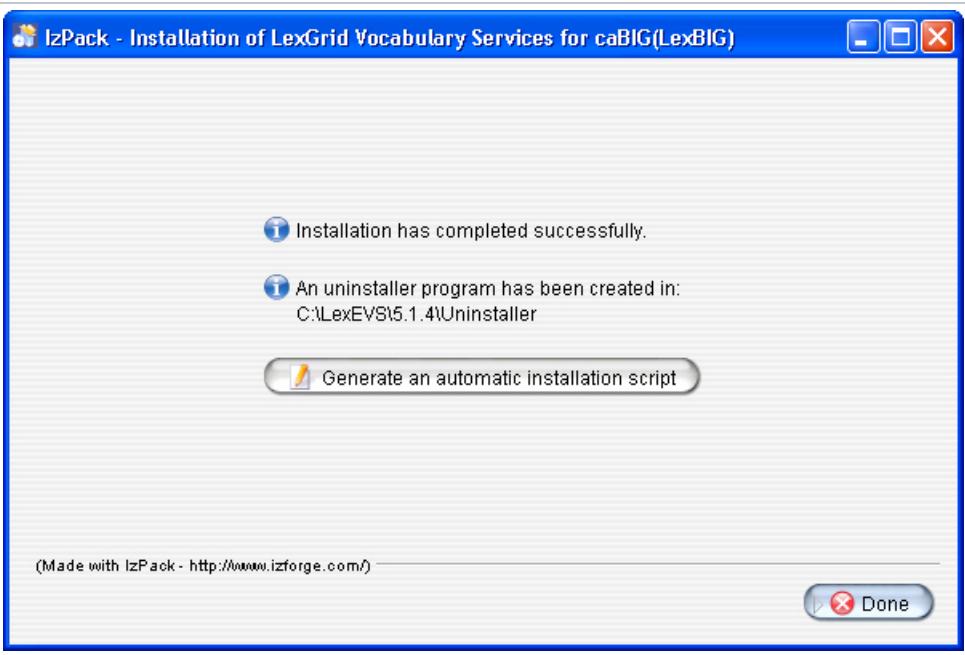
7. Select the components to be installed. The components "Product Information" and "Runtime and Administration Toolkit" are required and cannot be unchecked. The remaining components are optional. Clicking on individual components will display a description of that component. Once components have been selected click Next.



8. Once all the components have been installed, a Finished dialog box will be displayed. Click Next.



9. The last step of the installation wizard provides the ability to generate a command line installation configuration XML file that can be used on other servers. This installation script can be used to install LexEVS without the wizard that you just went through. If you want to generate this file then click the **Generate an automatic installation script** button. This will have no effect on the current installation. For more information look at the optional [#Downloading and Installing the LexEVS Local Runtime](#).



What's Inside

This section describes the location and organization of installation materials. Following installation, many of the following hierarchy of files and directories will be available (some features are optionally installable):

(As located in the LexEVS installation root directory)

Directory	Description of Content
/admin	Installed by default. This directory provides a centralized point for command line scripts that can be executed to perform administrative functions such as the loading, activation/deactivation, and removal of vocabulary resources. Object code used to carry out these functions is included directly in the LexEVS runtime components. Source code is included in the /source directory in the lbAdmin-src.jar (described below).
/doc	Optionally installed. This directory provides documentation related to LexEVS services, configuration, and execution. This guide is distributed in the /doc top-level directory.
/doc/javadoc	This directory provides documentation for model classes and public interfaces available to LexEVS programmers. Also included with each object representation is a UML-based model diagram that shows the object, its attributes and operations, and immediately linked objects. The diagrams work to provide clickable navigation through the javadoc materials
/examples	Optionally installed. This directory provides a small number of example programs. Refer to the README.txt file in this directory for instructions used to configure and run the example programs. The examples are intended to provide a limited interactive demonstration of LexEVS capabilities. Source and object code for the example programs is provided under the /examples/org subdirectory. Source materials are also centrally archived under the /source directory in the file lbExamples-src.jar.
/examples/resources	Contains sample vocabulary content for reference by the example programs; use the /examples/LoadSampleData command-line script to load for example code use or use one of the loaders in the /admin folder to load other content.
/gui	Optionally installed. This folder contains programs and supporting files to launch the LexEVS Graphical User Interface (GUI). The GUI provides convenient centralized access to administrative functions as well as support to test and exercise most of the LexEVS API. The GUI is launched using a platform-specific script file in the /gui directory. The name of the platform (e.g. Windows, OSX, etc) is included in the file name. Program source and related materials are centrally archived under the /source directory in the file lbGUI-src.jar.
/logs	Default location for log files, which can be modified by the LOG_FILE_LOCATION entry in the lbconfig.props file (see next section).
/resources	Installed by default. This directory contains resources referenced and written directly by the LexEVS runtime. It should, in general, be considered off-limits to modify or remove the content of this directory without specific guidance and reason to do so. Files typically stored to this location include the vocabulary registry (tracking certain metadata for installed content) and indexes used to facilitate query over the installed content. One file of particular interest in this directory is the /resources/config/lbconfig.props file. This file controls access to the database repository and other settings used to tune the LexEVS runtime behavior. Contents of this file should be set according to instructions provided by the <i>LexEVS Administrator's Guide</i> .

/runtime	<p>Installed by default. This directory contains a Java archive (.jar) file containing the combined object code of the LexEVS runtime, LexEVS administrative interfaces, and any additional code they are dependent on. All required code for execution of LexEVS administrative and runtime services is installed to this directory.</p> <ul style="list-style-type: none"> • /runtime/lbPatch.jar In the course of the product lifecycle, it is possible that smaller fixes will be introduced as a patch to the initially distributed runtime. Including this file in the classpath ensures automatic accessibility to the calling program without requiring adjustment. All patches are cumulative (there is at most one patch file introduced per release; all patch-level fixes are cumulative). • /runtime/lbRuntime.jar This is the standard runtime file, including all LexEVS and dependency code required for program execution except for SQL drivers (see next).
/runtime/sqldrivers	<p>The JDBC drivers used to connect to database repositories are not included in the lbRuntime.jar. Instead, the runtime scans this directory for the drivers to include. This can be overridden by path settings in the lbconfig.props file.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Note</p> <p>While the LexEVS software package ships with JDBC drivers to certain open source databases such as PostgreSQL, this folder provides a mechanism to introduce updated drivers or to add drivers which are license restricted for additional supported database systems.</p> </div> <p>For example, the Oracle database is supported by the runtime environment. However, the drivers are not redistributed with the LexEVS software. To run against Oracle, an administrator would add a jar with the appropriate JDBC driver to this directory and then reference it in the lbconfig.props settings. MySQL, the main test base for LexEVS, also requires a restricted license driver. Its driver can be downloaded here: mysql-connector-java-5.1.6.</p>
/runtime-components	<p>Optionally installed. Due to license considerations for additional materials (as described by the license.pdf and license.txt files in the install directory), the cumulative runtime provided in the lbRuntime.jar is not redistributable. This directory contains a finer grain breakdown of object code into logical components and 3rd party inclusions. All components are redistributable under their own license agreements, which are provided along with each archive. The top-level of the /runtime-components directory contains all code produced for the LexEVS project in the lexbig.jar.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Note</p> <p>These files are included as an alternative to the lbRuntime.jar for code execution and redistribution. There is no need to include any of these files in the Java classpath if you are already including the lbPatch.jar and lbRuntime.jar described above.</p> </div>
/runtime-components/extLib	This subdirectory includes all 3rd party code redistributed with the LexEVS runtime, along with respective license agreements.
/source	Optionally installed. This directory provides central accessibility to Java source for all code developed for the LexEVS project.
/test	Optionally installed. This directory provides an automated test bucket that can be used by System Administrators to verify node installation. Note that the /runtime/config/lbconfig.props file must still be configured for database access prior to invoking the test bucket. Testcases are launched via the TestRunner command-line script. Several reporting options are provided and are further described in the <i>LexEVS Administrator's Guide</i> .
/uninstaller	Contains an executable jar that can be invoked by an administrator to uninstall files originally introduced by the LexEVS installation.

Configuring the LexEVS Environment

The LexEVS Local Runtime can be configured to work with many different databases, however, the recommended databases are MySQL or PostgreSQL as noted in [LexEVS 5.x Prerequisite Software Supported](#). Following installation of a database of your choice you should follow these steps in order. The remaining text in this section is meant to help facilitate these steps.

1. Configure properties of your database software (MySQL and PostgreSQL examples shown below)
2. Create a database (commands for this are available in your database software documentation)
3. Obtain a JDBC driver for LexEVS to use when accessing the database.
 - Check {LEXEVS_HOME}\runtime\sqlDrivers to see if the appropriate driver for your database is there. Due to licensing reasons we can not ship some drivers, such as the MySQL driver.
 - MySQL driver can be downloaded from [Mysql driver mysql-connector 5.1.6](#).
4. Update the LexEVS lbconfig.props file (server properties) with connection and database settings

 **Note**

It is considered beyond the scope of these install instructions to address database setup and administration. However, proper database configuration is critical to the performance and long-term health of the LexEVS environment. System administrators should consult the MySQL or PostgreSQL documentation to determine settings that are appropriate to the host machine and environment. Tuning should be performed prior to loading vocabularies. The following tables provide settings that have been modified in database environments used during LexEVS development and adoption, and are provided for consideration by database administrators.

MySQL Properties

At a minimum you should consider changing these properties controlling the behavior of the MySQL database server. These properties are usually located in the file {MYSQL_HOME}/my.ini.

Property Name	Description
innodb_flush_log_at_trx_commit	Flush the transaction logs at each commit. Value: It is highly recommended that this option be set to '0' in Windows installations to improve load performance.
innodb_additional_mem_pool_size	Additional memory pool that is used by InnoDB to store metadata information. Value: 16M
innodb_buffer_pool_size	Buffer pool used to cache both indexes and row data. Value: 1G (consider going higher based on physical RAM available)
tmp_table_size	Maximum size for internal (in-memory) temporary tables. Value: 256M
query_cache_size	Query cache is used to cache SELECT results and later return them without actually executing the same query once again. Value: 64M
sort_buffer_size	This buffer is allocated when MySQL needs to rebuild the index in REPAIR, OPTIMZE, ALTER table statements as well as in LOAD DATA INFILE into an empty table. Value: 16M

Note

MySQL can be passed a JDBC option for the Windows local environment that may improve performance 30 to 50%.

Try the following values in the libconfig.props file for the DB_URL. This uses Windows Named Pipe function and avoids use of the TCP/IP protocol. It only works when connecting with a local iteration of the MySQL database on Windows.

```
DB_URL=jdbc:mysql://<dbname>?socketFactory=com.mysql.jdbc.NamedPipeSocketFactory
```

There are some commands that we have found useful in this environment. You can investigate this database software more at the [MySQL web site](#). If you have root or administrator access to your server you can change the password for the root user ID if you have forgotten or lost it.

```
mysqladmin -u root password \{mypassword\}
```

...where mypassword is your new password

Creating a database within the software can be done via the command line. The software does not come, by default, with GUI administration. That command might look like this:

```
mysqladmin -u root -p create \{DBNAME\}
```

...where DBNAME is the name of your database needed for LexEVS to store data in. You will be prompted for your password before the database is created.

PostgreSQL Properties - Modifying the postgresql.conf File for PostgreSQL

The following file contains properties controlling the behavior of the PostgreSQL database server:

```
PostgreSQL_HOME_DIRECTORY}/postgresql.conf
```

Property Name	Description
shared_buffers	Number of shared buffers. Value: 1000.
work_mem	The amount of memory in kilobytes allocated to working memory Value: 51200.
maintenance_work_mem	The amount of memory in kilobytes allocated to maintenance working memory. Value: 512000.
enable_seqscan	We set the enable_seqscan to false to use always use an index versus a table scan.

Server Properties

The next step is to configure LexEVS. This is wrapped up in editing one file, {LEXEVS_HOME}/resources/config/lbconfig.props. This file contains properties controlling the behavior of the LexEVS Local Runtime.

Note

The lbconfig.props file has the latest documentation embedded inside of it. lbconfig.props should be considered authoritative if there is a difference between the documentation written here and that contained in lbconfig.props.

You may open that file now and use the documentation within it to configure LexEVS and skip the rest of this section, keeping in mind that the information here is useful but not necessary to a successful install.

Below you will find an expanded overview of the options in this file. Options in **BOLD** must be modified so that LexEVS can properly use your database. They are also noted as required. There are many other options that you can change for performance reasons or alternative deployment scenarios, but you probably don't need to change them in a standard LexEVS installation. Some options can not be changed after you load terminologies into LexEVS. Now would be a good time to read about what is available to you.

Note

A single back slash '\' is not valid within the lbconfig.props file for file or directory paths (except within JDBC connection strings). When constructing file or directory paths you must use either a forward slash '/' or two backward slashes ' '.

Property Name	Description
DB_URL	Required: The address of your database server. This will include the host name or IP address followed by the name of the database that you created to store LexEVS data. For example: <pre>DB_URL=jdbc:mysql://{hostname}/{DBNAME}</pre> ...where hostname is the name of your server or host and DBNAME is the name of the database you created for LexEVS within your database software. hostname can also be an IP address hostname can also be followed by a colon and a port number in the case where you have multiple installs on the same server and have changed the port number that the database responds to. There are several options in the file as an example.
DB_PREFIX	Required for Oracle databases: The prefix to use on all tables or databases that LexEVS creates. If your database is Oracle, you may not use this feature. Leave the value equal to nothing. If you wish to run multiple LexEVS installations on the same database server, give them each a unique prefix. Do not use dashes '-' in the DB_PREFIX value. Recommended characters are alphanumeric (a-z, 0-98) and underscore '_'. The default is 'lb'.
DB_PARAM	Optional variable for passing extra database parameters. These will be appended to the end of the database connection string.
DB_DRIVER	Required: The Java class name that represents the driver that you wish to use with your database. There are several options in the file as an example.
DB_USER	Required: The database username. This account must have permission to add and remove tables, indexes, etc inside of this database. This account is typically created during the installation of the database software and is not necessarily a system user ID.
DB_PASSWORD	Required: The password for the database username.
LG_CONFIG_FILE	This is not actually a variable that you would set within this file. It is documented here for clarity of other variables that depend on this variable. Normally, this variable is automatically set (at runtime) to the location of the lbconfig.props file that is being used by the runtime. Alternatively, you can set the java system variable LG_CONFIG_FILE at system startup to point to the lbconfig.props file that you want LexEVS to use. Refer to additional documentation in the lbconfig.props file if you need to use this feature.
LG_BASE_PATH	This variable is the path that will be used to resolve any other relative (or unqualified) paths in the lbconfig.props file. This variable is optional, and should usually be left blank. If this variable is left blank, it will automatically be set (at runtime) to the location of the folder which contains the lbconfig.props file that the system was started with. This variable can also be overridden by setting the java system variable LG_BASE_PATH . The default value is blank.
JAR_FILE_LOCATION	The path of the folder that contains your SQL drivers and LexEVS extensions (if you have any). This value can be relative to the LG_BASE_PATH or absolute. The default is '..../runtime'
REGISTRY_FILE	The location of the file that will store information about all loaded terminologies. This value can be relative to the LG_BASE_PATH or absolute. The default is 'registry.xml'
INDEX_LOCATION	The folder where all LexEVS generated indexes will be stored. This folder can potentially be large (100+ GB) - depending on the terminologies loaded. This value can be relative to the LG_BASE_PATH or absolute. The default is '../lblIndex'

MAX_CONNECTIONS_PER_DB	LexEVS maintains a pool of connections to each database that it connects to. This sets the limit on the number of connections that will be opened. You may want to set this to a higher value - 20 or so (depending on expected user load) The default is 8.
CACHE_SIZE	LexEVS maintains an internal cache of some information that it needs to query from the database to resolve queries. This controls the size of that cache. This cache does not hold entire user queries. The default size is 500.
ITERATOR_IDLE_TIME	The length of time to allow an unused (and unclosed) iterator to stay valid before it is closed (and its resources freed) by the system. The default is 5 minutes.
MAX_RESULT_SIZE	This controls the maximum number of results that a user can resolve at one time for the CodedNodeSets and CodedNodeGraphs. This does not affect the total number of results that can be returned by an iterator, just the size that can be returned on the 'next' API call. The default is 1000.
LUCENE_MAX_CLAUSE_COUNT	The number of boolean 'clauses' Lucene will attempt to create for wildcard or Regular Expression queries. For very general queries on large ontologies, this will need to be set high. The default is 40000.
LOG_FILE_LOCATION	The path where LexEVS log files will be written. This value can be relative to the <code>LG_BASE_PATH</code> or absolute. The default is ' <code>../logs</code> '
API_LOG_ENABLED	Setting API call logging to 'true' will cause every method call to be printed to the log file. The default is 'false'.
SQL_LOG_ENABLED	Setting SQL logging to 'true' will cause every sql statement executed to be printed to the log file. The default is 'false'.
DEBUG_ENABLE	Setting debug to 'true' will give you more verbose logging information to debug problems. This should normally be set to 'false' since debug logging causes a negative performance impact. The default is 'false'.
LOG_CHANGE	Indicates when a new log file should be started. This can be set to 'monthly', 'weekly' or 'daily'. This can also be set to a number which will cause it to start a new log file whenever it reaches the specified number of MB in size. The default is 5 MB.
ERASE_LOGS_AFTER	If <code>LOG_CHANGE</code> is set to 'daily', 'weekly', or 'monthly', this variable instructs the logging service to remove log files that have not been written to in the specified amount of days. If <code>LOG_CHANGE</code> is set to a number of MB then this variable specifies the number of old log files that will be kept. The unit is treated as days regardless of the <code>LOG_CHANGE</code> value. Cleanup will only occur on restart of the JVM. The default is 5 files.
EMAIL_ERRORS	Used to enable or disable e-mail notification of system errors and warnings. If you set this to 'true', you must set <code>SMTP_SERVER</code> and <code>EMAIL_TO</code> . The default is 'false'.
SMTP_SERVER	The SMTP server to use to send errors over e-mail. Only applicable when <code>EMAIL_ERRORS</code> is set to 'true'. The default value is an example that must be modified to be useful.
EMAIL_TO	A comma separated list of e-mail address to set failure and warning notifications to. Only applicable when <code>EMAIL_ERRORS</code> is set to 'true'. The default value is an example that must be modified to be useful.

Testing the LexEVS Local Runtime

This Local Runtime installation provides a test suite to verify the environment. Before running these tests the Local Runtime and databases must already be configured.

Step	Action

1. In a command prompt window change to the test directory:

```
cd {LEXEVS_HOME}\test  
Run the TestRunner utility  
to start the test process:  
> TestRunner.{ext} -h
```

...where ext is the extension appropriate for your operating system (bat for Windows, sh for Unix).

For example:

- Windows:
TestRunner.bat -h
- Unix:
. ./TestRunner.sh -h

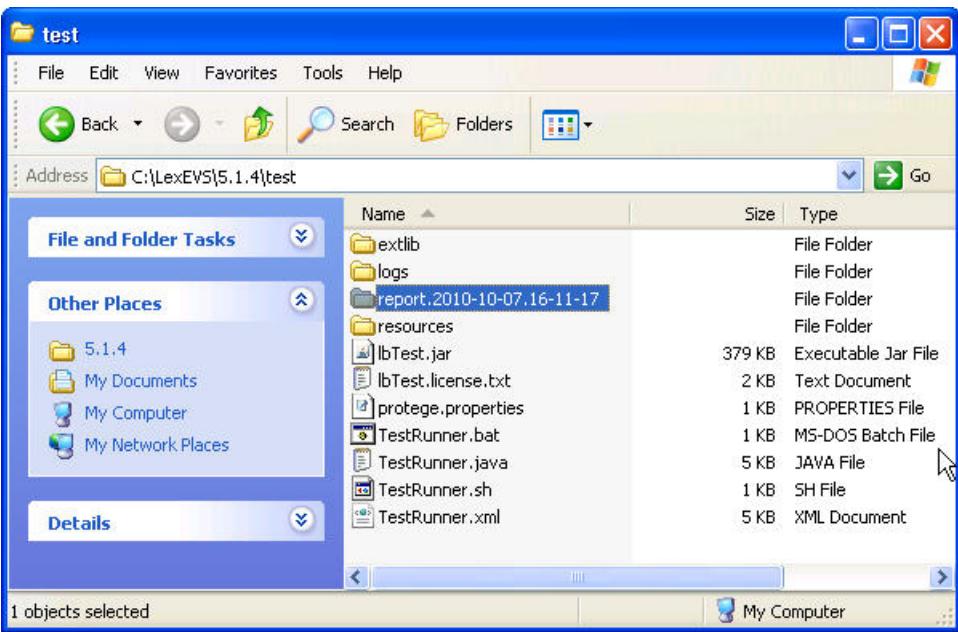


Note

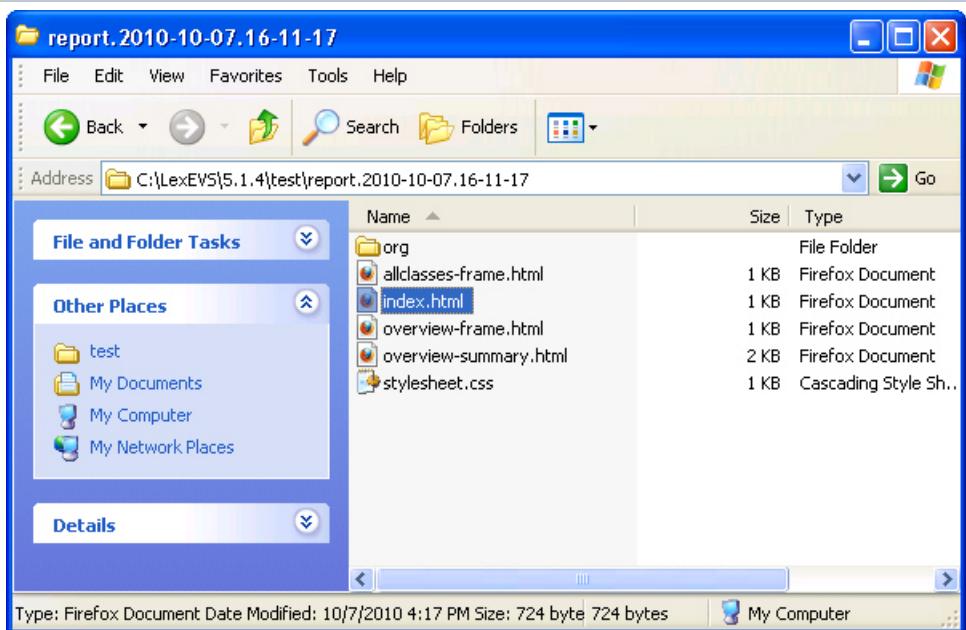
The image shows the test partially in progress.

```
C:\LexEVS\5.1.4\test>TestRunner.bat -h  
Unable to locate tools.jar. Expected to find it in C:\jre6\lib\tools.jar  
Buildfile: TestRunner.xml  
  
init:  
  
clean:  
    [mkdir] Created dir: C:\LexEVS\5.1.4\test\report.2010-10-06.14-48-15  
  
but-xml:  
    [junit] Running org.LexGrid.LexBIG.Impl.testUtility.AllTestsNormalConfig  
14:48:16,699 DEBUG LB_LOAD_LOGGER:395 - Freeing RadLexProtegeFramesLoaderImpl  
14:48:16,714 DEBUG LB_LOAD_LOGGER:395 - Freeing OBOLoaderImpl  
14:48:16,714 DEBUG LB_LOAD_LOGGER:395 - Freeing OWLLoaderImpl  
14:48:16,714 DEBUG LB_LOAD_LOGGER:395 - Freeing LexGridLoaderImpl  
14:48:17,027 DEBUG LB_LOAD_LOGGER:395 - Freeing LexGridLoaderImpl  
14:48:17,074 DEBUG LB_LOAD_LOGGER:395 - Freeing OWLLoaderImpl  
14:48:17,558 DEBUG LB_LOAD_LOGGER:395 - Freeing OWLLoaderImpl  
-
```

2. Use a file explorer to navigate into the directory that contains the test report. The report directory is placed in `LEXEVS_HOME`\test and is representative of the date and time you executed the test.



3. Launch the report into the default browser by double-clicking on the index.html file.



4. Review the test results. You should expect no failures or errors if the configuration of the Local Runtime and the databases is correct. If there are errors refer to the troubleshooting section.

Tests	Failures	Errors	Success rate	Time
800	0	0	100.00%	381.136

Name	Tests	Errors	Failures	Time(s)
org.LexGrid.LexBIG.Impl.testUtility	800	0	0	381.136

If all the tests pass then you have successfully installed the LexEVS Local Runtime. Congratulations!

Troubleshooting

- If you use a space in the name of the path to install LexEVS, that is `LEXEVS_HOME`, you might get errors from the underlying database during runtime. It is highly recommended that you do not use a space in the path.
- Upgrading LexEVS may require reloading content. Be sure to read the release notes for each LexEVS release before installing the latest version. Preserve configuration files and indexes unless instructed to reload or do otherwise in the release notes. These files include `lbconfig.props`, `registry.xml` and the entire `lbindx` directory in the resources directory. Note that the names of those last two files can be altered by variables in the `lbconfig.props` file.
- LexEVS loads of content are generally handled in a robust manner. Failed loads clean up after themselves relatively well when database management systems are properly configured to allow database drop functions by LexEVS. However, exiting the application in the middle of a load may cause unpredictable results in the databases, indexes, and lock files. They could be left in a state that will cause subsequent loads of the same terminology to fail.
 - Often these can be remedied by deleting the `lock.xml` file followed by using the cleanup function.
 - Other steps may need to be taken if this doesn't work, including dropping databases as the DBA, deleting the index file for the offending terminology, and editing the `registry.xml` and `metadata.xml` files by hand.
 - A quick, dirty solution is to drop all databases and delete all configuration files except `lbconfig.props`.

- Know when to scale database management systems for a large number of connections. If LexEVS is configured for multi-database loads and has multiple users connecting to all terminologies, then the administrator will need to scale database configurations to adapt to this. If you have a large number of terminologies loaded and a large user base connecting to the service using the lbGUI, then you will need the database configuration for number of connections scaled appropriately or users may not be able to connect.



Note

Loading in single database mode can eliminate this problem.

- LexEVS is no longer distributed with a Java MySQL driver due to licensing concerns. If LexEVS reports an error concerning establishing a connection to the MySQL server yet the MySQL CLI is able to connect, an adjustment in the version of Connector/J may be required.
 - The latest version of Connector/J is available from MySQL.org. The new jar should be placed in the following directory:

```
{LEXEVS_HOME}/runtime/sqlDrivers/
```

If you are changing drivers remove the existing driver jar file to ensure that the class loader does not incorrectly load the older driver file.

- If the user experiences slow performance with MySQL when loading or accessing terminologies then a review of the suggested configuration parameters for the database management system is recommended. Pay particular attention to the MySQL values.