

# LexEVS 6.x Local Runtime Configuration File Settings


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
CURRENT_PERSISTENCE_SCHEME	The current persistence scheme used to persist to an underlying store. This should remain unchanged unless there are custom persistence layers implemented. Default is set to 2.0
DB_URL	<p>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:</p> <pre>DB_URL=jdbc:mysql://{hostname}/{DBNAME}</pre> <p>...where <code>hostname</code> is the name of your server or <code>hostand</code> <code>DBNAME</code> is the name of the database you created for LexEVS within your database software <code>hostname</code> can also be an IP address <code>hostname</code> 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.</p>
DB_PREFIX	<p>The prefix to use on created tables.</p> <div><p> <b>Warning</b></p><p>If you are using Oracle, this is not supported because of length constraints on oracle table names so you must leave this value blank. On databases other than Oracle, there should be room for a reasonable (&lt; 10) amount of characters. This value <b>MUST BE UNIQUE</b> for each LexEVS runtime that uses the same database host. Do not use dashes - in the <code>db_prefix</code> value. Recommended characters are alphanumeric (a-z, 0-9) and underscore _. This variable should typically not be changed after content has been loaded.</p></div> <p>The default is <code>lb</code>.</p>
DB_PRIMARY_KEY_STRATEGY	<p>Primary key strategy indicates which strategy will be used for the primary key of the database tables.</p> <div><p> <b>Warning</b></p><p>This cannot be changed after the initial schema installation.</p></div> <p>Allowable values include:</p> <ul style="list-style-type: none"><li>• GUID Primary Keys are implemented as random GUIDs.</li><li>• <code>SEQUENTIAL_INTEGER</code> Primary Keys will be sequentially incremented as Integer values. Starting in 6.0.4 this is the default. Sequential Integer Primary Key (<code>SEQUENTIAL_INTEGER</code>) is the best strategy for large loads. This allows the database to insert records into the index in a sequential manner, which is more efficient. If the GUID strategy is used, records will be inserted into the index tree at random locations, resulting in index re-balancing after every insert.</li></ul>
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.

DB_PASSWORD_ENCRYPTED	Password encryption can be enabled by setting <code>DB_PASSWORD_ENCRYPTED</code> to <code>true</code> (case insensitive). Default and any value other than <code>true</code> is interpreted as <code>false</code> .
SINGLE_TABLE_MODE	<div>  <b>Warning</b> </div> <p><b>READ THE NEXT LINES CAREFULLY</b> <code>SINGLE_TABLE_MODE</code> indicates whether or not all loaded ontologies will be loaded to a single, common set of tables, or partitioned to one ontology per set of tables.</p> <ul style="list-style-type: none"> <li><code>true</code> - all ontologies in one common set of tables. Default is set to <code>true</code>.</li> <li><code>false</code> - each ontology in a separate set of tables. Setting single table mode to <code>true</code> may result in long delays during the removal of a large terminology. \\\</li> </ul>
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 <code>lbconfig.props</code> file that is being used by the runtime. Alternatively, you can set the java system variable <code>LG_CONFIG_FILE</code> at system startup to point to the <code>lbconfig.props</code> file that you want LexEVS to use. Refer to additional documentation in the <code>lbconfig.props</code> 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 <code>lbconfig.props</code> 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 <code>lbconfig.props</code> file that the system was started with. This variable can also be overridden by setting the java system variable <code>LG_BASE_PATH</code> . 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 <code>LG_BASE_PATH</code> or absolute. The default is <code>../../../../runtime</code>
REGISTRY_FILE	<i>This value is only here for backwards compatibility. It is not needed in versions beyond 6.x.</i> The location of the file that will store information about all loaded terminologies. This value can be relative to the <code>LG_BASE_PATH</code> or absolute. The default is <code>registry.xml</code> .
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 <code>LG_BASE_PATH</code> or absolute. The default is <code>../lbIndex</code> .
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 25.
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 <code>CodedNodeSets</code> and <code>CodedNodeGraphs</code> . 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.
LUCENEMAX_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 <code>true</code> will cause every method call to be printed to the log file. The default is <code>false</code> .
SQL_LOG_ENABLED	Setting SQL logging to <code>true</code> will cause every sql statement executed to be printed to the log file. The default is <code>false</code> .
DEBUG_ENABLED	Setting debug to <code>true</code> will give you more verbose logging information to debug problems. This should normally be set to <code>false</code> since debug logging causes a negative performance impact. The default is <code>false</code> .

LOG_CHANGE	Indicates when a new log file should be started. This can be set to <code>monthly</code> , <code>weekly</code> or <code>daily</code> . 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 LOG_CHANGE is set to <code>daily</code> , <code>weekly</code> , or <code>monthly</code> , this variable instructs the logging service to remove log files that have not been written to in the specified amount of days. If LOG_CHANGE 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 LOG_CHANGE value. Cleanup will only occur on restart of the JVM. The default is 5 files.