

# Performance Tuning

If you are running a production system where performance is critical and the database is large, you will eventually want to tweak your installation to improve tomcat and mysql performance.

## OpenMRS Settings

- Clear out the "patient.identifierRegex" [global property](#) to disable regular expression identifier searches.
- Clear out the "patient.identifierPrefix" and "patient.identifierSuffix" [global properties](#) to disable "like" identifier searches
- Make sure that the "dashboard.regimen.displayDrugSetIds" [global property](#) has concept ids, not names (e.g. "1085,1159" instead of "ANTIRETROVIRAL DRUGS,TUBERCULOSIS TREATMENT DRUGS")lba
- Set the 'searchWidget.batchSize', 'searchWidget.runInSerialMode' and 'searchWidget.searchDelayInterval' [global properties](#) to tune your searches for better performance and suit your implementation's environment e.g the speed of your network connection, typing skills and average number of simultaneous users on a typical work day. You might also consider reducing the value of the global properties person.searchMaxResults and increasing that of searchWidget.batchSize to reduce the load on the search widgets and server for better performance.
  - **searchWidget.batchSize** -- defines the maximum number of results fetched within a single request to the server. Defaults to 200. Smaller numbers will result in more, smaller exchanges of data; larger numbers will allow for fewer, larger exchanges of data. The ideal number will depend on your context (e.g., a system using satellite connections may work better sending fewer, larger packets of data).
  - **searchWidget.searchDelayInterval** -- is the number of milliseconds that the search widget wait after the last keystroke before searching. Defaults to 400 milliseconds. Smaller values can make searches feel quicker, but will create more network traffic from unnecessary requests.
  - **searchWidget.runInSerialMode** -- the search widget makes multiple queries to pull search results from the server. By default, the search widget will ask for everything it needs from the server at once (in parallel). If you are concerned about too much traffic coming to the server at once, you can change this setting to `true`. In most cases, this could slow down the return of results to the client; however, if your server is overloaded, then using serial mode and reducing the concurrent load on the server may actually help.
  - **searchWidget.maximumResults** - Specifies the maximum number of results to return from a single search in the search widgets, note that this differs from the batch size.
- When using MySQL with case insensitive collation set "search.caseSensitiveDatabaseStringComparison" Setting (formerly Global Property from 1.8 downwards) to "false". It will improve search performance on OpenMRS 1.7.4+, 1.8.4, 1.9.1+, 1.10+. See [TRUNK-2767](#).

## Tomcat

1. **Install tomcat from tomcat.apache.org (and don't use Ubuntu apt-get install).**

2. **Allowed Memory:** Three options for increasing Tomcat's allowed memory:

a. **At command line:** Start Tomcat with the parameters:

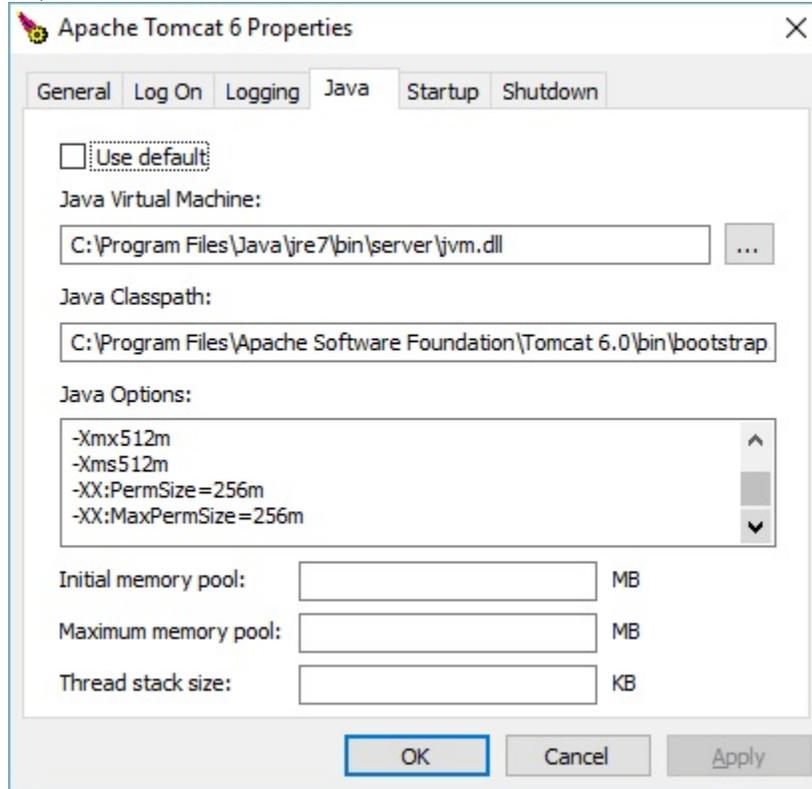
```
-Xmx512m -Xms512m -XX:PermSize=256m -XX:MaxPermSize=256m -XX:NewSize=128m
```

b. **if running Tomcat as a Windows Service:** open up the Tomcat Monitor (TOMCAT\_HOME/bin/tomcat5w.exe)

- Configure > Java > Java Options. Append:

```
-Xmx512m -Xms512m -XX:PermSize=256m -XX:MaxPermSize=256m -XX:NewSize=128m
```

Note: if you copy and paste the above into tomcat monitor, make sure that it doesn't insert an extra space at the end of each line, or tomcat will fail to start.



- c. if running Tomcat as a Linux daemon: open the /etc/init.d/Tomcat.sh script and add the following parameters to the CATALINA\_OPTS: If CATALINA\_OPTS is already set to this:
- CATALINA\_OPTS="-Djava.library.path=/opt/tomcat/lib/.libs" Then change it to this:

```
CATALINA_OPTS="-Djava.library.path=/opt/tomcat/lib/.libs -Xmx512m -Xms512m -XX:PermSize=256m -XX:MaxPermSize=256m -XX:NewSize=128m"
```

Note: When using FormEntry Module in a non-Windows environment (one that uses cabextract), do not adjust Tomcat memory usage to greater than 4GB. See [Ticket # FORM-14](#).

1. **Memory Leaks:** Tomcat has a default setting that causes memory leaks.
  - To turn it off, open the <TOMCAT HOME>/conf/web.xml file. In the jsp servlet definition add the following element:

```
<init-param>
  <param-name>enablePooling</param-name>
  <param-value>>false</param-value>
</init-param>
```

2. **Better Garbage Collection (Experimental):** As of JDK6\_u14, Sun introduce a better 'server' oriented [G1 garbage collector](#). This optimizes the garbage collection technique and can help reduce PermGen Out of Memory Exceptions. It is also recommended to use JDK6\_20 and higher as it has fixed bugs in G1 and improved performance greatly. To use G1, you need to add the following to CATALINA\_OPTS (as was shown above in [1]):

```
-XX:+UnlockExperimentalVMOptions -XX:+UseG1GC
```

## MySQL

1. Increase the **innodb\_buffer\_pool\_size**: It is the size in bytes of the memory buffer InnoDB uses to cache data and indexes of its tables. The larger you set this value, the less disk I/O is needed to access data in tables. On a dedicated database server, you may set this to up to 80% of the machine physical memory size. However, do not set it too large because competition for physical memory might cause paging in the operating system.
2. Increase the **max\_allowed\_packet** size: When MySQL gets a packet bigger than max\_allowed\_packet bytes, it issues a Packet too large error and closes the connection. Current real limit for the packet size is 1G, and practical limit is even lower.
  - Edit your INI file (e.g. for Windows, C:\Program Files\MySQL\MySQL Server x.x\my.ini).

- You must edit the actual INI file in use. MySQL comes with several alternative configurations (INI configurations). If you use the administration program, you should be able to find the `max_allowed_packet` setting — I found it using *MySQL Administrator* under the "Health" section's "System Variables" tab (*Connections > Networking > max\_allowed\_packet* in the hierarchy). (The details may differ between versions).
- On Linux, edit the `/etc/my.cnf` file
- Under the `[mysqld]` section, add these lines:

```
max_allowed_packet=64M
innodb_buffer_pool_size=3G
```

3. Run the [MySQL performance tuning script](#) and make adjustments to your MySQL configuration file based on the suggestions.
4. Note that you may also want to increase the `innodb_log_file_size`. I found that increasing this property to 25% of the `innodb_buffer_pool_size` greatly decreased the time it took to upgrade a database from Openmrs 1.6 to 1.9. However, note that you cannot simply change this parameter on an existing mysql install. Also, although it sped up the update process for me, it is still unclear to me if setting the log file size to be very high is good for overall performance. You can read more information here:
  - a. [http://www.mysqlperformanceblog.com/2011/07/09/how-to-change-innodb\\_log\\_file\\_size-safely/](http://www.mysqlperformanceblog.com/2011/07/09/how-to-change-innodb_log_file_size-safely/)
  - b. <http://www.mysqlperformanceblog.com/2008/11/21/how-to-calculate-a-good-innodb-log-file-size/>

## Installing the InnoDB plugin

To additionally improve the performance of mysql, you can install the InnoDB plugin. The plugin provides some performance improvements: in particular, fast index creation. When adding an index to a table, normally InnoDB needs to make a complete copy of the table. For tables like the `obs` table, which may have millions of rows, this can be very slow. Using the InnoDB plugin allows for creating indexes without requiring copying the entire table.

Information about installing the InnoDB plugin can be found here:

- Linux: <http://dev.mysql.com/doc/innodb-plugin/1.0/en/innodb-plugin-installation-dynamic-posix.html>

- Windows: <http://dev.mysql.com/doc/innodb-plugin/1.0/en/innodb-plugin-installation-dynamic-win.html>

I ran into some additional permission problems when installing the plugin that were not documented in the above notes. You should make sure that the user running the mysql daemon has access rights to read and execute the InnoDB plugin files (search for `ha_innodb_plugin` on your system). Also, on Ubuntu, apparmor was preventing mysql from accessing the plugin files. Information about fixing this problem can be found here: <http://ubuntuforums.org/showthread.php?p=8764880>