Troubleshooting Memory Errors

java.lang.OutOfMemoryError: PermGen space

This error will appear after Tomcat has exhausted all of its memory. This is typically caused by using the "Update" or "Reload" function on a webapp too many times. Currently, Tomcat and/or the JVM isn't releasing all of the memory when a webapp is destroyed/recreated. After several reloads Tomcat is out of its allotted memory and will hang or freeze.

Solutions

- Developers: Restart Tomcat after several updates/reloads
- Developers and Implementers: Allow Tomcat to use more memory. This will not stop the error, just prolong times between tomcat restarts. Follow the 'Java Heap Size' solution instructions.
- Implementers: Tell Tomcat to not leak memory

java.lang.OutOfMemoryError: PermGen space (Maven/Jetty)

This error will appear if Jetty (JVM) has exhausted all of its memory. This can occur on first time install of OpenMRS (during a DB create/upgrade process). To solve this problem one needs to increase the memory available to JVM (i.e. to Maven -- which invokes Jetty)

To solve the problem, set the following environment variable (MAVEN_OPTS) on your machine:

```
Increasing memory by setting env variable

export MAVEN_OPTS="-Xmx1024m -Xms1024m -XX:PermSize=256m -XX:MaxPermSize=512m"
```

Java Heap Size Exception

A Java Heap Size error occurs because of certain memory intensive actions (like Data Exporting). The new Spring managed API uses a lot of memory to cache the xml beans. You may need to increase the default memory allocation.

Solutions

Three options for increasing Tomcat's allowed memory:

- Start Tomcat with these parameters at the command line. Note: The current ANT task "Start Tomcat" will start with these options.

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

- If running Tomcat as a Windows Service, open up the Tomcat Monitor (TOMCAT_HOME/bin/tomcat5w.exe). Configure Tomcat -> Java -> Java Options, or go to Control Panel > Services > Apache Tomcat > Properties > Start Parameters. Append:

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

Note 1: 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.

Note 2: If running 64 bit version of Tomcat, you have to edit the Windows Registry HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Procrun 2.0\Tomcat5\Parameters\JavaJVM settings in Registry

Note 3: It no longer works to set these parameters through a JAVA_OPTS environment variable

- If running Tomcat as a Linux daemon, open the /etc/init.d/Tomcat.sh script and append the parameters to the CATALINA_OPTS variable: i.e.

```
If CATALINA_OPTS is already set to this: CATALINA_OPTS="-Djava.library.path=/opt/tomcat/lib/\lib" then change it to this: CATALINA_OPTS="-Djava.library.path=/opt/tomcat/lib/\lib -Xmx512m -Xms512m -XX:PermSize=256m -XX:
MaxPermSize=256m -XX:NewSize=128m"
```

For Ubuntu, see Jonah Mwogi's directions in the comments below.

Solution for Tomcat memory leak:

In <TOMCAT HOME>/conf/web.xml file, in the jsp servlet definition add the following element:
Finding Memory Solutions

OpenMRS developers use YourKit Profiler to discover and debug memory and cpu consumption.

YourKit is kindly supporting OpenMRS with its full-featured Java Profiler. YourKit, LLC is creator of innovative and intelligent tools for profiling Java and .NET applications. Take a look at YourKit's leading software products: YourKit Java Profiler and YourKit .NET Profiler.