

Blackboard Help

Learn / Administrator / Self- and Managed-Hosting / Release Notes / Individual Releases / 9.1 SP 10 / Performance and Scalability

Recommended JVM Tuning Set

For every Service Pack, the Blackboard Performance Engineering team spends many hours studying the transactional and systemic performance of Blackboard Learn and whether particular tuning sets will have a positive or negative impact on performance or scalability.

This section provides the recommended Java Virtual Machine (JVM) tuning set as executed in the Blackboard Performance Engineering laboratory against Release 9.1 Service Pack 10 and above. Customers are strongly encouraged to deploy their application environments in a 64-bit configuration using the recommended 4GB or larger tuning set.

Blackboard recommends using the following tuning set for 8GB JVMs on Windows, Linux, and Solaris systems with 4 CPUs using the latest version of Java 7. These recommended settings are managed in the `bb-config.properties` file.

```
bbconfig.min.heapsize.tomcat=8192m
bbconfig.max.heapsize.tomcat=8192m
bbconfig.max.permsize.tomcat=384m
bbconfig.max.stacksize.tomcat=384k (windows)
bbconfig.max.stacksize.tomcat=400k (Linux, solaris 32-bit)
bbconfig.max.stacksize.tomcat=800k (Solaris 64-bit)
bbconfig.jvm.options.platform=-XX:StackShadowPages=20 (Linux only)
bbconfig.jvm.options.platform=-XX:StackShadowPages=10 (Solaris only)
```



While the default `bbconfig.max.permsize.tomcat` setting is suitable for a default instance of Learn running the bundled set of Building Blocks it may be necessary to increase this setting when running a Learn installation with additional Building Blocks installed. Monitoring permgen usage using the Admin Console or similar tool of choice will provide optimal targets specific to your installation. Minimally consider an increase to 512m or if you have sufficient RAM available 768m to avoid out of memory issues.

```
bbconfig.jvm.options.extra.tomcat=-XX:NewSize=2048m -
XX:MaxNewSize=2048m -XX:+UseTLAB -XX:SurvivorRatio=4 -
XX:+AlwaysPreTouch -XX:-UseSplitVerifier -XX:+UseCompressedOops -
XX:+PrintVMOptions -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -
XX:+PrintGCTaskTimeStamps -XX:+PrintCommandLineFlags -
XX:+PrintGCApplicationStoppedTime -
XX:+PrintGCApplicationConcurrentTime
```

```
bbconfig.jvm.options.gc=-XX:+UseConcMarkSweepGC -XX:+UseParNewGC
-XX:ParallelCMSThreads=2 -XX:ParallelGCThreads=4 -
XX:CMSFullGCsBeforeCompaction=1
-XX:+CMSParallelRemarkEnabled
```

The below formulas are applied to derive the suggested tuning set. Please note that `NewSize` and `MaxNewSize` can take a range of values for optimal performance by dividing `bbconfig.max.heapsize.tomcat` property value by either 4 or 3.

```
-XX:NewSize = (bbconfig.max.heapsize.tomcat) / 4
-XX:MaxNewSize = (bbconfig.max.heapsize.tomcat) / 4
-XX:ParallelGCThreads = <#cpus < 8 ? #cpus : 3 + ((5 * #cpus) / 8) >
-XX:ParallelCMSThreads = (ParallelGCThreads + 3) / 4
```

A stack overflow in Java normally results in the offending thread throwing a `java.lang.StackOverflowError`. However, a limitation in Java implementation causes JVM to crash upon stack overflow error on some Unix servers running 64-bit JVM. Configuring the size of shadow pages in `bb-config.properties` addresses this issue. Please refer to [Java's Bug 7059899](#) for more detail.

```
bbconfig.jvm.options.platform=-XX:StackShadowPages=20 (Linux only)
bbconfig.jvm.options.platform=-XX:StackShadowPages=10 (Solaris only)
```

Blackboard Learn 9.1 Service Pack 10 and above require Java 7. This brings the opportunity to leverage Garbage-First (G1) garbage collector in Java 7 Update 4 and later. The G1 collector is a server-style garbage collector, targeted for multi-processor machines with large memories. Tests conducted by Blackboard Performance team indicated no significant performance gain on Blackboard Learn 9.1 Service Pack 10 and above. Additional experiments will be conducted G1 collector in the future releases.

