**1) JVM memory options related to java heap size**

Following three JVM options are used to specify initial and max heap size and thread stack size while running Java programs.

**-Xms**        set initial Java heap size

**-Xmx**        set maximum Java heap size

**-Xss**>         set java thread stack size

**2) JVM option to print gc details**

**-verbose:gc** logs garbage collector runs and how long they're taking. I generally use this as my first tool to investigate if GC is a bottleneck for a given application.

**-XX:+PrintGCDetails** includes the data from -verbose:gc but also adds information about the size of the new generation and more accurate timings.

**-XX:-PrintGCTimeStamps**  Print timestamps at garbage collection.

**3) JVM parameters to specify Java Garbage collector**

**-XX:+UseParallelGC**      Use parallel garbage collection for scavenges

**-XX:-UseConcMarkSweepGC** Use concurrent mark-sweep collection for the old generation. (Introduced in 1.4.1)

**-XX:-UseSerialGC**        Use serial garbage collection. (Introduced in 5.0.)  
beware when you use GC Parameters if you are working on time critical application e.g. high frequency trading application. As  GC is time consuming operation and its desired to create a balance.

**4) JVM debug options JVM options for remote debugging**

-Xdebug -Xnoagent -Xrunjdwp:transport=dt\_socket,server=y,suspend=n,address=8000  
to read more about remote debugging check [How to Setup Java remote debugging in Eclipse](http://javarevisited.blogspot.com/2011/02/how-to-setup-remote-debugging-in.html) and [10 Java debugging tips in Eclipse](http://javarevisited.blogspot.com/2011/07/java-debugging-tutorial-example-tips.html)

**5) JVM options related to profiling**

-Xprof

-Xrunhprof

**6) JVM options related to java classpath**

**Xbootclasspath** specifies classpath entries you want loaded without verification. The JVM verifies all classes it loads to ensure they don't try to dereference an object with an int, pop extra entries off the stack or push too many, and so on. This verification is part of the reason why the JVM is very stable, but it's also rather costly, and responsible for a large part of start up delay. Putting classes on the bootclasspath skips this cost, but should only be used when you know the classes have been verified many times before. In JRuby, this reduced startup time by half or more for a simple script. The -**Xbootclasspath** option can be used to either prepend (/p) or append (/a) resources to the bootstrap classpath. You Can read more about Java Classpath in my articles [How Classpath Works in Java](http://javarevisited.blogspot.com/2011/01/how-classpath-work-in-java.html) and [How to Solve ClassNotFoundException in Java](http://javarevisited.blogspot.com/2011/08/classnotfoundexception-in-java-example.html)

**7) JVM options to change  Perm Gen Size**

These JVM optiosn are quite useful to solve [java.lang.OutOfMemoryError:Perm Gen Space](http://javarevisited.blogspot.com/2011/09/javalangoutofmemoryerror-permgen-space.html).

-XX:PermSize and MaxPermSize

-XX:NewRatio=2  Ratio of new/old generation sizes.

-XX:MaxPermSize=64m     Size of the Permanent Generation.

**8) JVM parameters to trace classloading and unloading**

**-XX:+TraceClassLoading** and **-XX:+TraceClassUnloading** are two JVM options which we use to print logging information whenever classes loads into JVM or unloads from JVM. These JVM flags are extremely useful if you have any memory leak related to classloader and or suspecting that classes are not unloading or garbage collected.

**9) JVM switches related to logging**

**-XX:+TraceClassLoading** and **-XX:+TraceClassUnloading** print information class loads and unloads. Useful for investigating if you have a class leak or if old classes (like JITed Ruby methods in JRuby) are getting collected or not.

**-XX:+PrintCompilation** prints out the name of each Java method Hotspot decides to JIT compile. The list will usually show a bunch of core Java class methods initially, and then turn to methods in your application.

**10) JVM Switches for debugging purpose**

-XX:HeapDumpPath=./java\_pid.hprof  Path to directory or file name for heap dump.

-XX:-PrintConcurrentLocks       Print java.util.concurrent locks in Ctrl-Break thread dump.

-XX:-PrintCommandLineFlags   Print flags that appeared on the command line.