

# DrlvmCommandLineOptions

## Standard JRE Options

This content is now also available on our website [http://harmony.apache.org/cmd\\_options.html](http://harmony.apache.org/cmd_options.html)

To get the list of standard JRE options, run DRLVM with the `-help` option:

```
java -help
```

## Non-standard JRE Options

To get the list of non-standard JRE options, run DRLVM with the `-X` option:

```
java -X
```

## Component-specific Options and Properties

*(since revision r552672, (Jul 03, 2007))*

### VM Core (vm/vmcore)

OPTIONS{{ }}	COMMENTS
<code>-XX:-vm.assert_dialog</code>	Use to prevent assertion failures from popping up a dialog box.
<code>-XX:+vm.crash_handler</code>	Use to invoke gdb on crashes (POSIX only).
<code>-XX:-vm.finalize</code>	Use to disable finalization subsystem, that is, not to start finalization threads and to ignore finalization requests from the GC.
<code>-XX:+vm.jvmti_compiled_method_load_inlined</code>	Use to report inlined methods with <code>JVMTI_EVENT_COMPILED_METHOD_LOAD</code> event. Makes sense for optimizing jit.
<code>-XX:+vm.bootclasspath_appendclasspath</code>	Use to append classpath to the bootclasspath.
<code>-XX:gc.dll=&lt;GC_DLL&gt;</code>	Use to run DRLVM with non-default GC taken from <code>GC_DLL</code> dynamic library.

### EM (Execution Manager)

OPTIONS{{ }}	COMMENTS
<code>-Xverbose:em</code>	A commonly-used options to print method compilation events to <code>stdout</code> . Helps detect which method failed to compile (not an EM option, but very informative at EM behavior). This option is available in debug build only.
<code>-Xem:&lt;config&gt;</code>	Specify Execution Manager configuration file to use. There is a set of predefined configuration files ( <code>client</code> , <code>server</code> , <code>opt</code> , <code>jet</code> - please see below). You may also define your own EM configuration file.
<code>-Xem:client</code>	Use optimization sequence for client applications (default EM configuration).
<code>-Xem:server</code>	Use optimization sequence for server applications ('server' mode execution - analogous to <code>java -server</code> in HotSpot).
<code>-Xem:jet</code>	Use fast-compiling non-optimizing JIT only (Jitrino.JET).
<code>-Xem:opt</code>	Use aggressively optimising JIT compiler only (Jitrino.OPT).
<code>-XX:em.dll=&lt;dll&gt;</code>	Use to specify DRLVM-execution manager (EM) <code>dll</code> to load at startup.
<code>-XX:em.jitPath=&lt;path&gt;</code>	Use to override the default path for all JIT <code>dlls</code> in <code>EM.conf</code> with <code>path</code> .
<code>-XX:em.JIT_NAME.jitPath=&lt;path&gt;</code>	Use to override the default path for <code>JIT_NAME</code> JIT <code>dll</code> in <code>EM.conf</code> with <code>path</code> .
<code>-XX:em.properties=&lt;path&gt;</code>	Sets up a <code>path</code> to <code>em</code> configuration file to use.

### Jitrino

OPTIONS{{ }}	COMMENTS
<code>-XX:jit.arg.log=&lt;log_type&gt;</code>	This enables compilation logging for the Jitrino.OPT JIT compiler. <code>log_type</code> - a comma-separated list of the logging types to enable (see below the examples).
<code>-XX:jit.arg.log=rt</code>	Turnes ON all runtime logging.
<code>-XX:jit.arg.log=ct,dbg</code>	Turns ON compile-time logging and debug information.
<code>-XX:jit.arg.log=ct,irdump,dotdump,info</code>	Does a full compile-time logging for Jitrino ( <b>warning</b> : takes much disk space).
<code>-XX:jit.p.filter=.main -XX:jit.p.arg.log=ct,irdump,dotdump</code>	Enables a compile-time logging for methods having names started with <code>main</code> .
<code>-XX:jit.p.filter=java/lang/Math. -XX:jit.p.arg.log=ct,irdump,dotdump</code>	Enables logging for all methods of a single class.
<code>-XX:jit.arg.codegen.iprof=on</code>	Enables internal profiler dumps ( <code>iprof.stat</code> ) in Jitrino.OPT (needs <code>iprof.cfg</code> ).

## Thread Manager

OPTIONS{{ }}	COMMENTS
<code>-XX:+thread.soft_unreservation</code>	Turns ON soft lock unreservation.

## GCv5 (vm/gc\_gen)

OPTIONS{{ }}	COMMENTS
<code>-XX:gc.nos_size=N</code>	Specifies a fixed size for NOS (nursery object space or young generation).
<code>-XX:gc.num_collectors=N</code>	Specifies the number of collectors used for collections for parallel collection.
<code>-XX:+gc.use_large_page</code>	Use to turn on large page support of the platform.
<code>-XX:+gc.verify</code>	Use to turn on GC verifier for GC debugging, type to see available options.
<code>-XX:+gc.force_major_collect</code>	Use to force every collection to be full heap collection (i.e., no minor collections).
<code>-XX:+gc.ignore_finref</code>	Disables processing finalizer and weak references.
<code>-XX:-gc.heap_iteration</code>	Disables JVMTI functionality of heap iteration.

## GCv41 (vm/gc\_cc)

OPTIONS{{ }}	COMMENTS
<code>-verbose:gc</code>	Use to enable init messages from GC and information about GC pauses.
<code>-XX:-gc.adaptive</code>	Use to disable adaptive selection between GC algorithms: copying and sliding compaction.
<code>-XX:gc.type=N</code>	Use to specify the number of the GC algorithm to use (use together with <code>-Dgc.adaptive=0</code> ). Available algorithms: 1.copying + fallback to sliding compaction; 2.full collection using sliding compaction; 3.sliding compaction.
<code>-XX:+gc.heap_iteration</code>	Enables JVMTI functionality of heap iteration.
<code>-XX:+gc.ignore_finalizers</code>	Use to ignore finalizable objects, that is, not to track reachability of finalizable object, not to revive them and not to pass revived objects to VM for finalization.
<code>-XX:+gc.lp</code>	Use to turn on large page support on Windows to utilize TLB more efficiently. Use <code>-verbose:gc.lp</code> for more diagnostics.
<code>-XX:gc.lp=/mnt/huge</code>	Use on Linux to enable (1) or disable (0) using of large pages to utilize TLB more efficiently. Huge pages are allocated with the <code>1mmap1</code> method. You should specify the mount point where <code>hugetlbfs</code> is mounted. Use <code>-verbose:gc.lp</code> for more diagnostics.
<code>-XX:+gc.remember_roots_set</code>	Use to enable a root-set caching mode. By default, the copying collector starts evacuating objects as soon as it starts receiving roots during root set enumeration. This option disables immediate evacuation and ensures no objects are moved until the root-set enumeration is complete.

## BuildTest

COMMAND{{ }}	COMMENTS
<code>ant smoke.test -Dtest.mode=jit -Dtest.vargs="-Xem:server_static"</code>	Run all smoke tests in JIT <code>server_static</code> mode

```
ant smoke.test -Dtest.mode=jit -Dtest.vargs="-Xem:server_static" -Dtest.  
case=A
```

Run a single test case from smoke tests in JIT server\_static mode