

---

# Java Virtual Machine Option Reference for Mac OS X

Java



2009-03-04



Apple Inc.  
© 2009 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

.Mac is a registered service mark of Apple Inc.

Apple, the Apple logo, Carbon, Mac, Mac OS, WebObjects, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY,**

**MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

---

## **Introduction 5**

See Also 5  
Filing and Tracking Bugs 5

---

## **Java Virtual Machine Options 7**

A Note About Values 7  
General Options 7  
Mac OS X Specific 8  
Garbage Collection: Memory Usage 8  
Garbage Collection: General Settings 9  
Compilation 9  
Threading 9

---

## **Document Revision History 11**

---



# Introduction

---

This document describes the Java Virtual Machine runtime options that are specific to J2SE 5.0 and Java SE 6 in Mac OS X. It is written for Java developers who want to write Java applications for Mac OS X version 10.5. Although the audience is primarily developers of pure Java applications, it may also be useful for developers using Apple's web application framework WebObjects.

## See Also

For more information on Java Development on Mac OS X, read *Java Development Guide for Mac OS X*.

This document and other Java documentation for Mac OS X, including the Javadoc API reference, is available in the Java Reference Library. A subset of this documentation is installed in `/Developer/Documentation` on a Mac OS X system with the Xcode Tools installed. You can view this documentation through a web browser or through Xcode (from Xcode's Help menu, choose Documentation and then click Java).

General information on previous versions of Java for Mac OS X can be found in the Java Release Notes.

The main Apple website for Java technology, <http://developer.apple.com/java/>, contains links to information about Java development in Mac OS X.

The `java-dev` mailing list is a great source of information on a wide range of Java development topics in Mac OS X. You can sign up for this list at <http://lists.apple.com/mailman/listinfo/java-dev>.

Sun's Java web site, <http://java.sun.com/> is the essential reference point for Java development in general.

## Filing and Tracking Bugs

If you find issues with the implementation of Java that are not covered in this document or you want to follow the resolution of an issue, you may do so online through Radar, Apple's bug tracking system. To access Radar, you need an Apple Developer Connection (ADC) account. You can view the ADC membership options, including the free online membership, at <http://developer.apple.com/membership/>. With an ADC membership, you can file and view bugs at <http://bugreport.apple.com/>. When filing new bugs for Java in Mac OS X, please use `Java (new bugs)` for Component and `X as Version`.



# Java Virtual Machine Options

---

This document describes the less common and nonstandard VM options that are specific to Apple. All standard Java HotSpot VM options are available with the Mac OS X Java VM. Information on the standard HotSpot VM options can be found in the `java man` page. In addition to the standard options, many nonstandard (`-X` and `-XX`) options are available. Some of the more common nonstandard options are omitted here, but can be viewed by passing the `-X` option to `java`.

## A Note About Values

Except where noted, when bytes are specified as a parameter in an option, you may also designate kilobytes, megabytes, or gigabytes with the letters `k`, `m`, and `g`, respectively. You may use either uppercase or lowercase letters. For example, the following are all equivalent values: `4194304`, `4096k`, `4096K`, `4m`, and `4M`.

## General Options

`-server`

This option selects the HotSpot Server VM.

`-X`

Displays a brief description of the most common nonstandard VM options.

`-Xfuture`

Performs strict format-checking of class files. This option enforces a tighter conformance to the class file specification than the default, which is based on the standard in Java 1.1.x. You should test your code with this flag to ensure functionality in future versions of Java that may enforce stricter class file format-checking.

`-Xrs`

Same as `-XX:+ReduceSignalUsage`.

`-XX:-MaxFDLimit`

Directs the VM to refrain from setting the file descriptor limit to the default maximum. The default behavior is to set the limit to the value specified by `OPEN_MAX`, which is `10240`. Normally, this is the maximum number of files that a process may have open. It is possible, however, to increase this limit to a user-specified value with the `sysctl` utility. Under such circumstances, you may want to pass `-XX:-MaxFDLimit` to stop the Java VM from restricting the number of open files to `10240`.

`-XX:MaxDirectMemorySize=size in megabytes`

The maximum memory that can be allocated using NIO direct buffers. The default is `64 (64M)`.

`-XX:+ReduceSignalUsage`

Directs the VM to ignore `SIGHUP`, `SIGINT`, and `SIGTERM` signals. You need to implement handlers for these signals in native code as appropriate. Also, implement any relevant shutdown procedure from `System.exit`.

-XX:ReservedCodeCacheSize=*size in bytes*

Sets the maximum code cache size. The default value is 32M for the client compiler and 48M for the server compiler.

## Mac OS X Specific

-Xdock:icon=*pathToIconFile*

Sets the icon displayed in the Dock. By default, Mac OS X displays a generic Java icon unless you specify otherwise. You only need to use this for applications launched from the command line or from a JAR file. Double-clickable application bundles designate their icon in the `Info.plist` file.

-Xdock:name=*applicationName*

Sets the name to display in the Dock and in the menu bar. By default, Mac OS X displays the fully qualified name of the main class unless you specify otherwise. You need to use this only for applications launched from the command line or from a JAR file. Double-clickable application bundles get the appropriate name from the `Info.plist` file.

-XX:+UseFileLocking

Off by default, this option enables Carbon file locking. If your Java application will be interacting with files that may be simultaneously acted on by Carbon applications, use this option. It keeps the respective applications from modifying the file while the other is accessing it.

## Garbage Collection: Memory Usage

**Note:** Many of the garbage collection flags are dependent on the settings for the heap size. Make sure that you have the appropriate sizes set for the heap before fine-tuning how garbage collection uses that memory space.

-XX:MinHeapFreeRatio=*percentage as a whole number*

Modifies the minimum percentage of the heap kept free after garbage collection. The default value of 40. If at least 40% of the heap is not freed after garbage collection, the heap size increases.

-XX:MaxHeapFreeRatio=*percentage as a whole number*

Changes the maximum percentage of heap kept free after garbage collection before the heap is shrunk. The default value is 70. This means that if a garbage collection results in more than 70% of the heap being freed, the heap size decreases.

-XX:NewSize=*size in bytes*

Sets the default size for the Eden generation of allocated objects. The default value is 640K. (The `-server` flag increases the default size to 2M.)

-XX:MaxNewSize=*size in bytes*

Allows you to change the upper limit of the young object space in which new objects are allocated. The default value is 640K. (The `-server` flag increases the default size to 2M.)

-XX:NewRatio=*value*

Changes the ratio of new to old space sizes from the default value of 8 where the new space is 1/8 the size of the old space.



- XX:SurvivorRatio=*number*  
Modifies the ratio of size of the Eden to the survivor space from the default of 10 where Eden is survivorRatio+2 times larger than the survivor space.
- XX:TargetSurvivorRatio=*percentage*  
Desired percentage of survivor space used after scavenge. Default is 50.
- XX:MaxPermSize=*size in MB*  
Modifies the size of the permanent generation. The default is value is 32 (32MB).

## Garbage Collection: General Settings

- Xincgc  
This flag indicates that the VM should use concurrent mark and sweep garbage collection in incremental mode. It is equivalent to -XX:+UseConcMarkSweepGC -XX:+CMSIncrementalMode.
- XX:+UseConcMarkSweepGC  
Enables concurrent mark and sweep garbage collection. This option has an effect only on multiprocessor computers.
- XX:+UseParallelGC  
Enables parallel garbage collection. This option has an effect only on multiprocessor computers.
- XX:-DisableExplicitGC  
Ignores explicit calls to the System.gc method in your code. The VM still performs garbage collection when it normally would. This option just disallows you from explicitly forcing garbage collection in your code.
- XX:+PrintTenuringDistribution  
Prints tenuring age information for allocated objects in the young generation.

## Compilation

- XX:CompileThreshold=*value*  
Changes the number of method invocations (branches) before compilation begins. The default is 1500.
- XX:+CITime  
Displays how much time is spent in compiled code.
- XX:+PrintCompilation  
Prints a trace of the methods as they are compiled.

## Threading

- XX:NewSizeThreadIncrease=*size in KB*  
Specifies how much to increment the young object space size per active thread. This option may be useful in regulating an increased allocation rate due to increased threads. The default increment is 16 (16 kilobytes).

-XX:ThreadStackSize=*size in KB*

Changes the thread stack size from the operating system's default size.

-XX:+UseTLAB

Enables a thread-local allocation buffer so that heavily threaded applications can be allocated more efficiently, greatly increasing allocation performance. For Java 1.4.2, this option is on by default on multiprocessor computers and in Mac OS X Server. For J2SE 5.0, it is on for all configurations.

# Document Revision History

---

This table describes the changes to *Java Virtual Machine Option Reference for Mac OS X*.

Date	Notes
2009-03-04	First version as a separate document. Formerly part of "Java Property, VM Option, and Info.plist Key Reference for Mac OS X".

