
Java 1.4 System Properties

Cross Platform: Java



2005-04-29



Apple Inc.
© 2004, 2005 Apple Computer, 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

Apple, the Apple logo, Aqua, Mac, Mac OS, Panther, Quartz, 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 **Introduction to Java 1.4 System Properties** 5

Chapter 1 **Runtime System Properties** 7

Integration With the Native Application Environment 7

Rendering Hints 7

Full Screen Java 8

Graphics Drawing Performance 9

Pixel Conversion Performance 10

Rendering Performance 10

Window Positioning 12

Document Revision History 13

Introduction to Java 1.4 System Properties

This document lists the runtime system properties supported in Java 1.4.2. Most of the properties are also supported in Java 1.4.1. The properties are grouped according to their functionality. Note that these properties may not all be supported in future versions of Java in Mac OS X.

If you find issues with the implementation, please report them at <http://bugreport.apple.com>. When filing new bugs, use Java (new bugs) for Component and X as Version.

INTRODUCTION

Introduction to Java 1.4 System Properties

Runtime System Properties

Most of these properties accept a Boolean value. If they accept a different value, that is noted. You can call them from the command line with the `-D` flag to `java` or in your code with `System.setProperty`. For example, you could set your Swing application to use the Mac OS X menu bar with either

```
java -Dapple.laf.useScreenMenuBar="true" yourApplication
```

or

```
System.setProperty("apple.laf.useScreenMenuBar", "true");
```

Integration With the Native Application Environment

`apple.laf.useScreenMenuBar`

If you are using the Aqua look and feel, this property puts Swing menus in the Mac OS X menu bar. Note that `JMenuBar`s in `JDialog`s are not moved to the Mac OS X menu bar.

The default value is `false`. Java applications created with Xcode have this property set to `true`.

`apple.awt.brushMetalLook`

Allows you to display your main windows with the “textured” Aqua window appearance. This property should be applied only to the primary application window, and should not affect supporting windows like dialogs or preference windows.

The default value is `false`.

`apple.awt.fileDialogForDirectories`

By default, the AWT File Dialog lets you choose a file. Under certain circumstances, however, it may be proper for you to choose a directory instead. If that is the case, set this property to allow for directory selection in a file dialog.

The default value is `false`.

`apple.awt.showGrowBox`

Most native Mac OS X windows have a resize control in the bottom-right corner. By default, Java application windows that use the Aqua look and feel show this control, but there may be circumstances where you want it to be hidden. This property is used to decide if the grow box is shown or not.

The default value is `true`.

Rendering Hints

`apple.awt.antialiasing`

Causes graphic primitives like line, arc, rectangle, and so on, to be painted with antialiasing. By default, text also takes this setting, though you can override it using `apple.awt.textAntialiasing`. Even

with this property set to `on` from the command line, you may still set the `KEY_ANTIALIASING` rendering hint for specific objects.

This option accepts either `on` or `off` for its value. Although this property is `off` by default, it is set to `on` when you use the Aqua look and feel. This makes the behavior more consistent with the native Mac OS X user interface. Note that even if you set this to `off` for an application that uses the Aqua look and feel, Aqua user interface elements themselves are still drawn with antialiasing.

`apple.awt.textantialiasing`

Sets the `KEY_TEXT_ANTIALIASING` rendering hint. Although this property inherits the same setting as `apple.awt.antialiasing`, you can override that setting explicitly.

This option accepts either `on` or `off` for its value. The default value is `off` unless you are using the Aqua look and feel.

`apple.awt.rendering`

Determines whether Graphics2D objects prioritize speed or quality. This option accepts either `speed` or `quality` for its value. It sets the `KEY_RENDERING` hint so that it accepts either `VALUE_RENDER_SPEED` or `VALUE_RENDER_QUALITY` as an argument.

`apple.awt.interpolation`

Sets the `KEY_INTERPOLATION` rendering hint to determine which algorithm is used in image transformations. This option accepts either `nearestneighbor`, `bilinear`, or `bicubic` for its value. Setting this option passes `VALUE_INTERPOLATION_NEAREST_NEIGHBOR`, `VALUE_INTERPOLATION_BILINEAR`, or `VALUE_INTERPOLATION_BICUBIC`, respectively, to `KEY_INTERPOLATION`.

`apple.awt.fractionalmetrics`

Sets the `KEY_FRACTIONALMETRICS` to use floating point font metrics instead of the default integer metrics. Options include `on` and `off` to set `VALUE_FRACTIONALMETRICS_ON` or `VALUE_FRACTIONALMETRICS_OFF`.

Full Screen Java

`apple.awt.fakefullscreen`

Causes full screen applications to be displayed in a window. You might want to use this property during development of full screen Java applications.

The default value is `false`.

`apple.awt.fullscreencapturealldisplays`

When you have multiple displays, entering full-screen mode normally darkens the secondary screens. Setting this property to `false` overrides the default behavior and secondary screens are not darkened. You might want to override the default behavior for development purposes like debugging.

The default value is `true`.

`apple.awt.fullscreenhidecursor`

Hides the mouse cursor when in full-screen mode.

The default value is `true`.

`apple.awt.fullscreenusefade`

If you change the screen resolution when entering full-screen mode, the screen transitions by fading out of the old resolution and back in with the new resolution. If you do not change screen resolution,

you normally do not see this fade effect. This property enables that fade effect regardless of whether you have changed the screen resolution.

The default value is `false`.

Graphics Drawing Performance

`apple.awt.graphics.OptimizeShapes`

By default, Java in Mac OS X tries to use graphics primitives in place of the relatively more complex shape objects if there is an appropriate mapping. For example, a call to draw a simple shapes like `draw(new Rectangle2D.Float(0, 0, 10, 10))` is mapped to `drawRect(0, 0, 10, 10)`. If you do not want this automatic optimization, set this value to `false`.

The default value is `true`.

`apple.awt.graphics.EnableLazyDrawing`

By default, drawing actions for graphics primitives are queued before being sent to the renderer. This keeps the renderer primed and improves the graphics performance for rendering simple primitives like lines, rectangles, arcs, and ovals. This is referred to as lazy drawing. You may disable lazy drawing by setting this value to `false`.

The default value is `true`.

`apple.awt.graphics.EnableLazyDrawingQueueSize`

If lazy drawing optimization is enabled, this option sets the size of the queue used. This property takes an integer value which represents the number of graphics primitives cached. Each graphics primitive requires about 10 entries. Each entry requires 4 bytes (1 entry = 4 bytes).

The default value is `10`.

`apple.awt.graphics.EnableQ2DX`

Quartz 2D acceleration is a feature included in Mac OS X v.10.4 for development purposes. It uses hardware acceleration to speed up rendering of simple primitives like images, lines, rects, and simple characters. In addition to using this flag, you need to enable Quartz 2D acceleration in the Quartz Debug application, included with the Xcode Developer Tools for Mac OS X v.10.4.

This is strictly a developer option. Java applications intended for use on Mac OS X v.10.4 should not rely on the presence of Quartz 2D acceleration.

The default value is `false`.

`apple.awt.graphics.EnableDeferredUpdates`

Mac OS X v.10.4 uses deferred drawing updates, which eliminates visual tearing, but blocks those applications that flush too often. Deferred drawing updates are not supported for Java applications. If you want to enable deferred drawing throughout your application, use this system property.

This is strictly a developer option. Java applications intended for use on Mac OS X v.10.4 should not rely on deferred drawing updates.

The default value is `false`.

Pixel Conversion Performance

`apple.awt.graphics.EnableLazyPixelConversion`

By default, the Java implementation in Mac OS X optimizes pixel conversion for image formats that are not natively supported by the underlying operating system. Image formats that are not supported natively by Core Graphics include:

- `TYPE_3BYTE_BGR`
- `TYPE_4BYTE_ABGR`
- `TYPE_4BYTE_ABGR_PRE`
- `TYPE_BYTE_BINARY`
- `TYPE_BYTE_INDEXED`
- `TYPE_CUSTOM`
- `TYPE_INT_ARGB`
- `TYPE_INT_BGR`
- `TYPE_USHORT_565_RGB`
- `TYPE_USHORT_GRAY`

You may override the default behavior by setting this property to `false`.

The default value is `true`.

`apple.awt.graphics.UseTwoImageLazyPixelConversion`

Optimize pixel conversion of `BufferedImage.TYPE_INT_ARGB` images by using either Java or native based pixels as needed. (This is only used when the image is the source.)

The default value is `true`.

Rendering Performance

The following properties all allow you to turn off specific aspects of rendering. These are useful if you are debugging performance issues and need to pinpoint where a particular performance issue is occurring in your code. The default setting for all of them is `true`, where the indicated item is rendered. Setting any of these properties to `false` will cause that particular item not to be rendered.

For example, if you notice a significant slowdown in your program and suspect that it might be occurring when you the renderer is filling rectangles, you could pass in

`apple.awt.graphics.RenderFillRect=false` and note whether that significantly speeds up the performance of your code.

`apple.awt.graphics.RenderLine`

Determines whether lines are rendered.

The default value is `true`.

`apple.awt.graphics.RenderDrawRect`

Determines whether rectangles are drawn by the renderer.

The default value is `true`.

- `apple.awt.graphics.RenderFillRect`
Determines whether rectangles are filled by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderDrawRoundRect`
Determines whether round rectangles are drawn by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderFillRoundRect`
Determines whether round rectangles are filled by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderDrawOval`
Determines whether ovals are drawn by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderFillOval`
Determines whether ovals are filled by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderDrawArc`
Determines whether arcs are drawn by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderFillArc`
Determines whether arcs are filled by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderDrawPolygon`
Determines whether polygons are drawn by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderFillPolygon`
Determines whether polygons are filled by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderDrawShape`
Determines whether shapes are drawn by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderFillShape`
Determines whether shapes are filled by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderImage`
Determines whether any images are drawn by the renderer.
The default value is `true`.
- `apple.awt.graphics.RenderString`
Determines whether strings are rendered.
The default value is `true`.
- `apple.awt.graphics.RenderGlyphs`
Determines whether glyphs are rendered.
The default value is `true`.

`apple.awt.graphics.RenderUnicodes`

Determines whether Unicode text is rendered.

The default value is `true`.

Window Positioning

`apple.awt.window.position.forceSafeCreation`

Enforces the creation of new windows on screen. New windows are not created offscreen where users would not be able to access them.

The default value is `false`.

`apple.awt.window.position.forceSafeProgrammaticPositioning`

Prohibits windows from being moved programatically into a position where users are unable to access them. The `true` setting promotes optimal interaction between the Java environment and the native window server. Setting this property to `false` may result in unpredictable behavior in the windowing environment.

The default value is `true`.

`apple.awt.window.position.forceSafeUserPositioning`

Prohibits users from moving windows into a position where they would no longer be able to access them.

The default value is `false`.

Document Revision History

This table describes the changes to *Java 1.4 System Properties*.

Date	Notes
2005-04-29	Added definitions for <code>apple.awt.graphics.EnableQ2DX</code> and <code>apple.awt.graphics.EnableDeferredDrawing</code> .
2004-10-05	Added property and description for <code>apple.awt.fileDialogForDirectories</code> . Revised description for <code>apple.awt.showGrowBox</code> .
2004-01-08	Changed title to <i>Java 1.4 System Properties</i> .
	Removed <code>com.apple.mrj.application.apple.menu.about.name</code> system property.
	Added brief introductory text.
2003-09-20	Minor structural and grammatical changes. Includes information relevant to Java 1.4.1 Update 1.
2003-06-20	Includes information new system properties present in the Developer Preview release of Java 1.4.1 for Mac OS X version 10.2 as well as in Mac OS X Panther.
2003-06-11	Released as a standalone document.
2003-05-15	Released as an appendix to <i>Java 1.4.1 Development for Mac OS X</i> .

REVISION HISTORY

Document Revision History