
Java for Mac OS X v10.5 Update 1 Release Notes

Cross Platform: Java



2008-05-02



Apple Inc.
© 2008 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

Apple, the Apple logo, AppleScript, Aqua, Mac, Mac OS, Objective-C, Quartz, and Safari are trademarks of Apple Inc., registered in the United States and other countries.

Intel and Intel Core are registered trademarks of Intel Corporation or its subsidiaries 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 on Mac OS X v10.5 Update 1 Release Notes** 5

Who Should Read This Document? 5

Organization of This Document 6

Chapter 1 **Known and Resolved Issues** 7

Activity Monitor 7

Java Application Support 7

Java Aqua LAF 8

Java AWT 11

Java Classes 13

Java HotSpot 13

Java JavaLib 14

Java JDK 14

Java Networking 14

Java Sound 15

Java Swing 15

Java Text 15

Java Toolkit 16

Java Web Start 16

Shark 16

Document Revision History 19

Introduction to Java on Mac OS X v10.5 Update 1 Release Notes

These release notes explain Java on Mac OS X v10.5 Update 1 and some known and resolved issues that may affect developers creating Java applications for Mac OS X v10.5 Update 1.

Java on Mac OS X v10.5 Update 1

Java for Mac OS X v10.5 Update 1 provides Java SE 6 version 1.6.0_05 to all 64-bit capable Intel Macs. This release is only for Mac OS X v10.5.2 and later, and should not be installed on earlier versions of Mac OS X. This release provides several additional features including support for AppleScript as a `javax.script` language, new API for altering the application Dock icon, and native support for document modal sheets.

Who Should Read This Document?

Any developer who wants to develop Java applications on Mac OS X v10.5 should read this document as various issues and fixes found in this release may affect your application. Java developers should also read this document for the most current information on Java SE 6 and 64-bit support in Mac OS X v10.5.

In addition, this document contains new details regarding:

AppleScript as a supported language to the `javax.script` API.

New API in `com.apple.eawt.Application`:

- Get and set the Application Dock icon
- Set a badge label on the Dock icon
- Set the Dock menu
- Open the native system help viewer if your application has a help bundle

New API in `com.apple.eio.FileManager`:

- `getPathToApplicationBundle()` obtains the path to the currently running Java application package

Document Modal Sheets can be created by setting the the `"apple.awt.documentModalSheet"` client property to `Boolean.TRUE` on the `JRootPane` of `JDialogs`.

`GraphicsConfiguration.getNormalizingTransform()` now returns a transform value that represents the physical size of the attached monitor.

The system focus ring color can now be obtained using: `UIManager.getColor("Focus.color");`

Aqua-style title-less group borders can now be obtained using:

`UIManager.getBorder("InsetBorder.aquaVariant");`

Java DTrace probes are supported.

Java SE 6 can be used as a supported Java VM for Applets in 64-bit capable browsers. Java SE 5 will continue to be used in 32-bit-only capable browsers, such as Safari.

Organization of This Document

This document contains the following chapter:

- [“Known and Resolved Issues”](#) (page 7) highlights a selection of high-visibility bugs that have been addressed in this release. This chapter is broken down by the category where the bug occurs and provides a brief description of what the issue was and how it was resolved.

This document also contains a revision history.

Known and Resolved Issues

This chapter lists high visibility bugs that have been addressed in this release. It is not a complete listing of all of the bugs addressed. If you still have issues with any of these bugs, please file a new bug at <http://bugreport.apple.com/> under the Java (new bugs) component, version X. Refer to the bug number indicated below in your new bug if you believe it is the same issue.

Activity Monitor

Radar #5782402

Java applications show up as (null) in Activity Monitor

Description:

Bundled java applications running in a 64-bit JVM show up as "(null)" in Activity Monitor.

Workaround:

Use the command line "ps" utility to see the full name of the application.

Java Application Support

Radar #4561714

Bundled applications cannot display a Java 6 splash screen

Description:

New in Java 6 is the ability to display a splash screen when your application starts. From the command line, this is done with the '-splash:<path>' argument. No such equivalent existed for bundled applications.

Resolution:

Fixed. Bundled applications can now display a splash file by adding the key 'SplashFile' with the path to the image file as the corresponding value to your bundled application's Java dictionary. This feature only applies to applications running in Java 6.

Radar #5150621

Can't select Java SE 6 in Java Preferences.app

Description:

Java SE 6 is not visible in the Java Application Runtime Settings list in Java Preferences.app.

Resolution:

Fixed. Java 6 can now be selected as your primary Java VM for use on the command line and for Java applications and Web Start applications that request "1.4+" or "1.5+" as their VM version.

Radar #5422774

Running Java applications in 64-bit mode.

Description:

Java 1.5 on Mac OS X 10.5 will run in 64-bit mode, but the command line 'java' will use the 32-bit Java VM by default.

Resolution:

To specify the architecture you want, use '-d32' or '-d64' to indicate which Intel architecture you want to use. If the specified architecture is not supported the default architecture will still be used.

Java 6 handles both -d32/-d64 on the command line, but always runs in 64-bit mode. J2SE 1.4 does not handle -d32/-d64 on the command line (identifies them as unrecognized options that cause the JVM not to be started).

Radar #5753008

Prioritizing the J2SE 5.0 (64-bit) JVM actually selects the 32-bit J2SE 5.0 JVM.

Description:

The 32-bit J2SE 5.0 JVM is actually selected and used by some Java and Web Start applications when J2SE 5.0 (64-bit) has the highest priority in Java Preferences' Java Application Runtime Settings.

Radar #5787297

Preferred Java VMs are reset after installing Java for Mac OS X v10.5 Update 1.

Description:

The preferred order of Java VMs for launching Java applications is reset after installing Java for Mac OS X v10.5 Update 1. This information is now also stored in ~/Library/Preferences.

Workaround:

Open /Applications/Utilities/Java/Java Preferences.app, change the preferred VM's, and click "Save".

Java Aqua LAF

Radar #4777837

JTabbedPane throw ArrayIndexOutOfBoundsException when calling removeAll()

Description:

JTabbedPane with enough tabs to scroll between them could sometimes throw an `ArrayIndexOutOfBoundsException` originating from the Aqua Look and Feel if `remove()` or `removeAll()` was called on the component.

Resolution:

This issue has been addressed.

Radar #5445548

JOptionPane dialogs in Web Start Applications use the generic Java application icon if not launched from a bundled application.

Description:

JOptionPanes use the icon of the application to show which application the modal dialog belongs to. Java Web Start applications launched directly from a JNLP file would not show their icon as specified in the JNLP until they were saved as a bundled application, and relaunched.

Resolution:

JOptionPanes now obtain the application icon directly from the Dock tile of the application.

Radar #5599191

"JComboBox.isPopDown" and "JComboBox.isSquare" don't repaint the component when changed.

Description:

The extended JComboBox client properties described in Technical Note 2196 did not update the visual appearance of the JComboBox until some external event triggered a repaint.

Resolution:

Changing the JComboBox client properties now requests a `repaint()` and updates the JComboBox immediately.

Radar #5600749

Client properties ignored if set too early

Description:

If client properties were set while a non-Aqua look and feel was active, they were ignored when switching to the Aqua Look and Feel.

Resolution:

Fixed. Client properties are now correctly processed when the Aqua Look and Feel is installed.

Radar #5601325

Up/Down arrow keys should jump to the beginning or end of line

Description:

Multiline JTextComponents don't move the insertion caret to the beginning of the first line when the up-key is pressed and the insertion point is already in the middle of the first line of text. Same problem with down-key pressed and the last line.

Resolution:

The insertion caret now jumps to the beginning or end of line using the up/down arrow keys, as per the standard multi-line text field behavior in Mac OS X.

Radar #5618485

The client property to specify a descending sort order for a JTable column is misspelled.

Description:

The descending value for the "JTableHeader.sortDirection" client property key was mistakenly misspelled "decending".

Resolution:

Accepted values of "JTableHeader.sortDirection" are now { "ascending", "decending", "descending", null }.

Radar #5683832

No way to obtain a title-less Mac OS X style group border

Description:

Mac OS X style titled borders can be obtained using `UIManager.getBorder("TitledBorder.aquaVariant")`, however there was no way to obtain a Mac OS X style title-less border.

Resolution:

Title-less Mac OS X style group borders can now be obtained with `UIManager.getBorder("InsetBorder.aquaVariant")`.

Radar #5686535

There is no way to access the color of the focus ring from Java

Description:

While the light and dark selection colors are available from the `UIManager`, there is no way to obtain the color of the focus ring from a Java application.

Resolution:

The focus ring color can now be obtained using `UIManager.getColor("Focus.color")`.

Radar #5693868

The vertical alignment of icons positioned in square-style JButtons is 2 pixels too high

Description:

JButtons set to the square style, or resized vertically to become compressed into the square style did not adjust the vertical alignment of their icons to compensate, and was 2 pixels too high.

Resolution:

JButtons icons now are center aligned in all button styles.

Radar #5696011

Bevel-style JButtons could overflow their visual bounds

Description:

JButtons with non-center alignment could place their text and icon content on top of the visual border of the button.

Resolution:

All button styles (particularly bevel), now compensate for their visual margin when laying out their internal content. Edge-aligned content should now rest against the edge, and not paint on top of it.

Java AWT

Radar #5493967

GraphicsEnvironment.getScreenDevices() returns inaccurate information

Description:

The GraphicsDevice and GraphicsConfiguration returned inaccurate or incomplete information for available accelerated memory and physical screen size.

Resolution:

GraphicsDevice.getAvailableAcceleratedMemory() now returns the available texture memory on the specified display device's graphics card, if available. GraphicsConfiguration.getNormalizingTransform() now returns a non-identity transform that represents the scaling proportion between the real physical size of the attached display and the virtual coordinate space of the Java application.

Radar #5512101

Java applications launched from the command line show 'bin' as their name in the Dock.

Description:

Some Java command-line tools displayed 'bin' as the name of the application in the Dock.

Resolution:

Fixed. The Dock now shows the correct application name for Java applications and Java command-line tools.

Radar #5600385

Java application windows don't work with Spaces

Description:

Switching between spaces and using Command-Tab to return to a Java application, did not switch back to the Java application's space. Menus appeared, but no windows were visible.

Resolution:

Fixed. Java SE 6 windows now correctly interact with Spaces.

Radar #5664937

Copying and pasting images to and from Java fails

Description:

Copy and paste operations on image types from Java applications fail with a native exception logged to the console.

Resolution:

This issue has been resolved.

Radar #5674015

Mac OS X does not have the concept of a "System Tray"

Description:

Java applications using the SystemTray API in Java SE 6 need a visual representation on Mac OS X.

Resolution:

Traylcons are represented on Mac OS X using NSStatusMenus that are presented to the left of the standard system menu extras. The `java.awt.Image` artwork for a Traylcon is presented in grayscale as per the Mac OS X standard for menu extras. `Traylcon.displayMessage()` presents a small non-modal dialog positioned under the Traylcon. The `ActionListener` for the Traylcon is only fired if the "OK" button on the non-modal dialog is pressed, and not if the window is closed using the window close button. Multiple calls to `Traylcon.displayMessage()` will dismiss prior messages and leave only the last message. If the application is not in the foreground when `Traylcon.displayMessage()` is called, the application bounces its icon in the Dock. Message windows are badged with the application's icon to identify the which application triggered the notification.

Radar #5686843

The system property `awt.multiClickInterval` returns null.

Description:

`System.getProperty("awt.multiClickInterval")` always returned null.

Resolution:

Java now returns an integer representing the number of milliseconds of the double-click delay time based on the user's setting for "Double-Click Speed" in the Keyboard & Mouse preference panel.

Radar #5754483

Command-line Java apps cannot run graphically without a Dock icon

Description:

Java applications started from a command line could not be started in UIElement mode.

Resolution:

Set the Java system property "apple.awt.UIElement" to "true" to start your command-line application in UIElement mode. This suppresses creation of a Dock icon and menu bar, and is only appropriate for modeless transient utility applications.

Radar #5762408

Swing JTextComponents do not interact with the Mac OS X Services menu

Description:

It is not possible to perform operations from the Services menu on lightweight text components.

Resolution:

All Java-accessible text components that support the concept of selection now interact with the Mac OS X Services menu under the application menu, including JTextComponents. Services can now request the current selection, and replace the current selection.

Java Classes

Radar #4743709

AppleScript should be available as a language to the javax.script API

Description:

AppleScript should be provided as a ScriptEngine to the javax.script API in Java SE 6.

Resolution:

AppleScript has been provided as an available language to the javax.script API. It supports passing basic Java objects and collections into AppleScript functions, and receiving Java objects as return values from AppleScript scripts. Currently supported types include strings, numbers, images, lists, sets, and maps/records. The AppleScriptEngine cannot be instantiated in applications that do not have privileges to perform Runtime.exec() operations.

Java HotSpot

Radar #4643752

Java process instrumentation tools don't work

Description:

Java process instrumentation tools like jinfo, jmap, and jstat, don't work without escalated privileges.

Workaround:

Use the associated tools with the 'sudo' command.

Java JavaLib

Radar #5569300

Thread.getContextClassLoader() returns null with -XstartOnFirstThread

Description:

Thread.getContextClassLoader() will return null if the application is run from Eclipse, or with -XstartOnFirstThread specified on the command line.

Resolution:

Thread.getContextClassLoader() now returns the correct class loader for the current thread whether the application was started with -XstartOnFirstThread or not.

Java JDK

Radar #5655357

Passing -client to the Java Virtual Machine does not enable the Client HotSpot compiler

Description:

The Client HotSpot VM compiler is not available for the 64-bit Java SE 6 for Mac OS X.

Java Networking

Radar #4623491

No IPv6 addresses returned for the local machine

Description:

Calling InetAddress.getAllByName() with the local machine hostname as the argument would return only IPv4 addresses. It would not return any IPv6 addresses bound to the host.

Resolution:

This problem has been addressed. The method now returns both IPv4 and IPv6 addresses.

Java Sound

Radar #5616768

javax.sound.sampled.spi.MixerProvider has incorrect information

Description:

The file javax.sound.sampled.spi.MixerProvider referred to audio providers not available on Mac OS X.

Resolution:

Fixed. Non-functional providers are no longer offered.

Java Swing

Radar #4877367

In-window menu bars do not show Mac OS X menu shortcut symbols

Description:

JMenus inside of JFrames show menu shortcuts as "Meta" and "Alt" instead of using the "Command" or "Option" symbol.

Resolution:

KeyEvent.getKeyModifiersText() and KeyEvent.getKeyText() now return the unicode symbols as presented in the screen menu bar on Mac OS X (Command, Option, Control, Shift, Numeric Keypad, etc).

Java Text

Radar #5599927

Text rendered using the Quartz pipeline (including printing) is distorted

Description:

Text rendered using the Quartz graphics pipeline (including all printing) did not apply the font face or metrics appropriately for scaled graphics contexts, resulting in irregularly spaced and untransformed characters.

Resolution:

This issue has been resolved.

Java Toolkit

Radar #5420927

There is no way to set a custom Dock icon or menu from Java.

Description:

There has been no way to set a custom Dock icon, set a Dock icon badge, or add a custom popup menu to a Java application's Dock icon.

Resolution:

New API has been added to the `com.apple.eawt.Application` object to support setting and getting the Dock icon image, setting a custom red badge on the icon (like Mail's unread count), and setting and getting a custom popup menu that is prepended to the standard Dock menu items.

Radar #5680303

Can't reference a Java application's bundle from Java.

Description:

Currently, Java applications need native code to get the path to their application bundle.

Resolution:

`com.apple.eio.FileManager.getPathToApplicationBundle()` now returns the full path to the application. This is the equivalent of calling "`[[NSBundle mainBundle] bundlePath]`" from Objective-C.

Java Web Start

Radar #3834108

Web Start applications don't display splash screens

Description:

Web Start ignored the request for a splash screen when an icon kind of 'splash' was specified in the JNLP file.

Resolution:

Fixed. Web Start now displays any requested splash screen for any version of Java.

Shark

Radar #4989222

Can't profile Java SE 6 applications with Shark

CHAPTER 1

Known and Resolved Issues

Description:

Java SE 6 applications cannot be profiled with Shark. This is because libShark.jnilib is 32-bit only.

Workaround:

None available at this time.

CHAPTER 1

Known and Resolved Issues

Document Revision History

This table describes the changes to *Java for Mac OS X v10.5 Update 1 Release Notes*.

Date	Notes
2008-05-02	First version of Java for Mac OS X v10.5 Update 1 Release Notes.

REVISION HISTORY

Document Revision History