
Java for Mac OS X v10.5 Release Notes

Cross Platform: Java



2007-10-31



Apple Inc.
© 2007 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, Aqua, Bonjour, Carbon, Cocoa, FireWire, Keychain, Leopard, Mac, Mac OS, Quartz, Safari, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

Finder is a trademark of Apple Inc.

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 Release Notes** 5

Who Should Read This Document? 5

Organization of This Document 5

Chapter 1 **Known and Resolved Issues** 7

Java Accessibility 7

Java Applets 7

Java Application Support 8

Java Aqua LAF 9

Java AWT 15

Java Documentation (developer) 16

Java Engine 17

Java Events 18

Java Graphics 18

Java HotSpot 19

Java InputMethods 20

Java JavaLib 20

Java JDK 21

Java PB Templates 22

Java Printing 22

Java Security 23

Java Sound 23

Java SWT Support 23

Java Text 24

Java Toolkit 25

Java Tools 25

Java Web Start 26

Document Revision History 27

Introduction to Java on Mac OS X v10.5

Release Notes

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

Java on Mac OS X v10.5

Java on Mac OS X v10.5 is comprised of J2SE 5.0 (1.5.0_13-b05) and J2SE 1.4 (1.4.2_16-b05). J2SE 1.3 is not present on Mac OS X v10.5. J2SE 5.0 is recommended for all development and deployment on Mac OS X, and all use of J2SE 1.4 is deprecated.

J2SE 5.0 on Mac OS X v10.5 contains many new features including 64-bit Intel support, a refreshed Swing Aqua Look and Feel, improved SWT interoperability, and a Sun 2D graphics pipeline.

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 new features and outstanding issues with Java on Mac OS X v10.5.

In addition, this document contains new details regarding:

- Deploying your Mac OS X bundled and WebStart applications for 64-bit.
- Deploying your SWT/RCP-based applications via WebStart.
- Tips for enhancing your Swing applications for the Mac OS X Aqua experience.

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.

INTRODUCTION

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

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.

Java Accessibility

Radar #3939296

JScrollPane contents were not visible to VoiceOver

Description:

The contents of JScrollPane are now traversable by VoiceOver.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Java Applets

Radar #4190188

Safari hangs if an applet specifies an 'image' but the image can't be found

Description:

Safari became stuck with a spinning beach ball when an applet specified an 'image' as part of its parameter list, but the image file could not be found on the server.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #5068385

Can't detect Java version from a web browser

Description:

Prior to Mac OS X 10.5, the Java plugin reported its version through the plugin's description attribute in JavaScript. The MIME types reported by the plugin were incorrect.

Resolution:

The currently selected version of Java used for applets now reports MIME type information consistent with Sun's Java plugin for other platforms via the `navigator.mimeTypes` array. You can look for 'application/x-java-applet;version=x.x' for all major versions of Java up to 1.5 or 1.4.2, and 'application/x-java-applet;jpi-version=1.5.0_13' or 'application/x-java-applet;jpi-version=1.4.2_16', depending on the selected version of Java.

Java Application Support

Radar #3428173

Help books in bundled Java applications are not registered with the Help viewer

Description:

In prior versions of Java, a Java application could bundle a help book in the application and it would be automatically registered with the Help viewer. Starting in Java 1.4.1, this functionality was lost.

Resolution:

Java applications with bundled help books are now automatically registered with the help viewer if the Java property "apple.awt.useSystemHelp" is set to "true". If you were using the workaround provided in the HelpHook code sample, it will continue to work, but is no longer needed as of Mac OS X 10.5.

Radar #3597977

The application menu in Java applications was not localized for many languages

Description:

Java is now localized into all of Mac OS X's standard languages.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4471286

Bundled Java applications cannot request a 64-bit Java VM

Description:

Bundled Java applications use a 32-bit Java VM by default, even on 64-bit-capable hardware.

Resolution:

Java applications can specify which architecture will be used for the application. To do this, add an array in the Java dictionary of the application's Info.plist named "JVMArchs". Each element in the array indicates the preferred hardware architectures you want your application to use. If the first choice is not available, the second will be used, and so on. You may also specify per-architecture settings for the Java dictionary keys "VMOptions" and "ClassPath". Use a key of "VMOptions.<architecture>" to specify Java VM options that should only be used for the specified architecture. Use a key of "ClassPath.<architecture>" to specify a class path

that should be used for the specified architecture. The per-architecture values are set in addition to those specified in the non-architecture-specific key, and will override the non-architecture-specific values, if applicable.

Radar #4581633

JavaApplicationStub is not backward compatible

Description:

Old stubs are ppc-only, and do not work on 10.3.9 when QT7.1 is installed. New stubs do not work on older OS versions.

Resolution:

The current JavaApplicationStub works on ppc and intel and is backward compatible to 10.1.5. Note that for code developed in Xcode 3.0 to run on Java 1.3.1 references to com.apple.eawt must be removed and the property apple.laf.useScreenMenuBar will not work since both were introduced in Java 1.4. Furthermore, Java application bundles created by Xcode 3.0 are missing the PkgInfo file and appear as directories in Finder on 10.1.5. Copying a PkgInfo file from a bundle created by Jar Bundler remedies this.

Java Aqua LAF

Radar #3479922

JToolBar buttons did not have a pressed icon

Description:

JToolBar buttons now have a Mac OS X toolbar style, and icons on each JButton now have a pressed and focused icon synthesized for them.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #3560238

There has been no way to create a circular indeterminate progress indicator in Java.

Description:

It is now possible to create a circular indeterminate progress indicator from a JProgressBar in Swing. Setting the "JProgressBar.style" client property to "circular" will create a small round indicator when the progress bar has been setIndeterminate(true). Please see tech note TN2196 "New Control Styles available within J2SE 5.0 on Mac OS X 10.5" for more information.

Resolution:

This problem is now fixed.

Radar #3723348

Bi-directional text did not render correctly in JTextComponents and the insertion point indicator did not match up with the actual insertion point

Description:

The Sun2D text renderer now contains appropriate shaping hints to handle bidi text, and is now the default renderer on Java for Mac OS X.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4518714

JTooltips did not match the native platform appearance, and appeared underneath certain UI components (the window resize indicator)

Description:

JTooltips were conditionally implemented as lightweight or heavyweight popup windows depending on their location and intersection with parent windows. Now in Leopard, Java implements all popup windows as heavyweight windows. These heavyweight windows have shadows when the Aqua look and feel is installed, and have a flat appearance when other look and feels are installed (ex: Metal or GTK).

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4518717

JTable headers did not match the native Mac OS X appearance (centered alignment)

Description:

JTable headers are now leading (usually left) aligned in the Aqua look and feel. The default alignment in all Sun-provided look and feels is centered.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4518743

JSliders did not match the native Mac OS X slider behavior

Description:

JSliders now snap-to-ticks as the thumb is dragged, instead of smoothly dragging and snapping after the mouse is released. Single clicks on JSlider tracks now set the thumb value directly under the mouse click instead of incrementing the thumb towards the click point. This behavior varies from all other Sun provided look and feels.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4631846

Horizontal mouse wheel scrolling did not work in JScrollPanes

Description:

Horizontal scroll events are now delivered to JScrollPanes as Shift+ScrollWheel events, since there is no horizontal scrolling API in Java. Horizontal scrolling mouse events will now move the content view horizontally, along with Shift key modifier events.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4681685

JSliders did not honor the "Slider.paintThumbArrowShape" client property

Description:

The "Slider.paintThumbArrowShape" client property is used by the Metal look and feel to indicate that the slider should show an arrow thumb instead of a round thumb. The Aqua look and feel now honors this property as well.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4802357

JTextFields did not match the native disabled Mac OS X text field appearance

Description:

JTextFields now have a visually distinct border color when they are disabled.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4841927

There has been no way to specify smaller versions of the default Mac OS X Aqua controls

Description:

Many JComponents now change their default size, appearance, and font when the "JComponent.sizeVariant" client property is set to "small" or "mini". Please see the tech note "New Control Styles available within J2SE 5.0 on Mac OS X 10.5" for more information.

Resolution:

This problem is now fixed.

Radar #4858198

JOptionPanels did not match the native Mac OS X dialog appearance

Description:

Default info JOptionPanels now use the application's icon, as per the Aqua User Interface Guidelines instead of a message bubble icon. Warning and error JOptionPanels now show the application's icon badged onto the warning or error icon as per the Aqua User Interface Guidelines.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4890361

JTextFields did not match the native focus/selection behavior of Mac OS X

Description:

JTextFields now select their contents when they gain focus through keyboard tab cycling. If the insertion point is in the middle of the text, the insertion point will remain and the entire text will not be selected.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4901662

JComboBoxes could not select items using the keyboard

Description:

JComboBoxes now show popups, move the selection, and commit the selection using the up, down, and enter keys as expected.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #4907470

There is a new implementation of the Aqua Look and Feel in Mac OS X Leopard. All references to the apple.laf.* package are deprecated.

Description:

The Aqua Look and Feel in Mac OS X Leopard is a new implementation, and does not provide backwards compatibility to any client code that uses classes in the apple.laf.* package. The legacy Aqua Look and Feel classes are present in this release for Mac OS X, but will be disabled and/or removed in future releases of Java on Mac OS X.

Workaround:

Use the values installed in the UIManager to obtain system colors, icons, and other features of the native platform appearance. There are also numerous new client properties in Java for Mac OS X Leopard to modify existing standard controls. Please file bugs at <http://bugreport.apple.com> to request additional features not provided by the Aqua Look and Feel.

Radar #5040323

JPasswordField did not match the native password field appearance in Mac OS X.

Description:

JPasswordField now use the bullet character instead of asterisks as the default echo character. This is different from all other Sun provided look and feels.

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #5042764

Accelerator mnemonic keys are not consistent with the Aqua user interface

Description:

Java no longer paints the underlined mnemonic character for components that have an accelerator key set until the user holds down the "option" key. The same accelerator behavior is still present, but the underline is not drawn on menus or controls until "option" key is held down by the user.

Resolution:

This problem is now fixed.

Radar #5043653

JTextAreas did not match the native selected appearance of text regions in Mac OS X.

Description:

JTextAreas now retain a light grey selection region when they are not focused.

Resolution:

This problem is now fixed.

Radar #5046173

JTextFields had clipped focus rings.

Description:

JTextFields now show a full focus ring around their exteriors when they are laid out using their preferred size. JTextFields that are vertically constrained smaller than their preferred height will progressively clip their focus rings and maintain their horizontal insets. JTextFields that are setOpaque(true) will clip their focus ring on all sides, and will have no exterior padding.

Resolution:

This problem is now fixed.

Radar #5053990

There existed no way to create a search field in Java resembling the search fields in Mac OS X.

Description:

JTextFields can now be converted into search fields using the "TextField.variant" client property set to "search". Please see the tech note "New Control Styles available within J2SE 5.0 on Mac OS X 10.5" for more information and additional options to configure JTextField based search fields.

Resolution:

This problem is now fixed.

Radar #5060060

The SystemColor color instances did not reflect the actual color values used in Mac OS X.

Description:

The SystemColor colors have been updated, and several now update live based on the selection in the Appearance Preference Pane (example: SystemColor.textHighlight).

Resolution:

This problem is now fixed in Mac OS X 10.5.

Radar #5060393

Titled borders do not look like the Mac OS X group box appearance

Description:

Titled borders cannot be set to the Mac OS X group box appearance because all the items inside of the bordered component would have to be setOpaque(false) to match the group box background color. This behavior can be elected into voluntarily though using a value pre-populated into the UIManager.

Workaround:

To create a single titled border with the Mac OS X group box style:

```
Border border = UIManager.getBorder("TitledBorder.aquaVariant");
if (border == null) border = BorderFactory.createEtchedBorder();
BorderFactory.createTitledBorder(border, "Title");
```

To use the Mac OS X group box style on all titled borders:

```
Border border = UIManager.getBorder("TitledBorder.aquaVariant");
if (border != null) UIManager.put("TitledBorder.border", border);
```

Radar #5221572

JScrollBars become unusable when shrunk

Description:

JScrollBars now remove their arrows when shrunk smaller than 64 points. The thumb is still available for moving the content until the size of the scroll view becomes smaller than the minimum size of the thumb.

Resolution:

This problem is now fixed.

Radar #5298465

Swing JOptionPane dialogs do not show the current application icon

Description:

As the Aqua Human Interface Guidelines prescribe, every application on Mac OS X should show its application icon in modal dialogs requiring the user's attention. Java now provides this support in Swing JOptionPane icons. Information, Question, and Plain dialogs that used the "info" icon now use the bundled application's icon instead. Warning and Error dialogs now also have the warning or error icon, with the application's icon superimposed on top of it.

Resolution:

This problem is now fixed.

Java AWT

Radar #2568391

Cocoa NSViews cannot host a java.awt.Frame

Description:

Java on Mac OS X has never had support for embedding a java.awt.Frame in a Cocoa NSView.

Resolution:

Mac OS X Leopard offers a new framework called JavaFrameEmbedding.framework that provides view-level support for embedding Java frames in a window. See the headers in that framework or the sample code for JavaFrameEmbedding for more information.

Radar #4364631

Java windows should be able to opt into the textured appearance individually

Description:

There are now a number of per-window client properties that can be set including palette style titlebars, window opacity, and textured background. See tech note TN2196: "New Control Styles available within J2SE 5.0 on Mac OS X 10.5" for more information.

Resolution:

This problem is now fixed.

Radar #5015972

Windows set to full screen mode with the "apple.awt.fakefullscreen" system property did not hide the menu bar or dock

Description:

Full screen windows now hide the menu bar and dock when the "apple.awt.fakefullscreen" system property is set.

Resolution:

This problem is now fixed.

Radar #5045905

There is no way to create a Java window with a document "proxy icon".

Description:

There is a new client property ("Window.documentFile") that can be added to the JRootPane of each heavyweight window that takes a File object as a key. When this client property is set, the window will then show a document "proxy icon" next to its title. Please see tech note TN2196 for more information.

Resolution:

This problem is now fixed.

Radar #5362292

Drag-and-drop of files to a Java application fails if the file name contains non-Roman characters

Description:

Files with non-Roman characters could not be passed to java.io.File objects.

Resolution:

Java now pre-composes all filenames of files dragged to Java applications or chosen via a JFileChooser. Files created from File.list() and File.canonicalize() are also normalized to pre-composed unicode.

Java Documentation (developer)

Radar #4451855

BigDecimal Backward Compatibility

Description:

In J2SE 5.0, BigDecimal.toString() returns a value using scientific notation instead of the value as a string, the standard behavior in Java 1.4.2.

Workaround:

Calling BigDecimal.toPlainString() returns the value as a string.

Radar #5226690

Bundled applications cannot run in 64-bit mode

Description:

Bundled Java applications deployed on 64-bit Intel machines cannot be forced to use a particular architecture by clicking a checkbox in the Get Info panel for the application.

Resolution:

If you want your bundled Java application to run in 64-bit you need to make the following changes to the Java dictionary in the Info.plist file. -- Set the value of "JVMVersion" to 1.5* or 1.5+. -- Add an array element named "JVMArchs". Each of the elements must be of type String, and specify the order of preference for architectures. The available architectures of the machine running the application are compared against this list, and the first one that matches is chosen. For example, `<key>JVMArchs</key> <array> <string>x86_64</string> <string>i386</string> <string>ppc</string> </array>` would choose 64-bit Intel if available, then i386. If the application is run on Power PC hardware, it will run as it did before Mac OS X 10.5. If you specify a list and none of your choices could not be satisfied, the application will not launch.

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 Engine

Radar #4960744

JDI returns incorrect bytecodes for wide version of ALOAD and ASTORE opcodes

Description:

Fixed JDI to return correct bytecode.

Resolution:

This problem is now fixed.

Radar #5317088

gdb crashes while trying to attach to a Java application

Description:

gdb crashes while you are trying to attach to a Java application.

Workaround:

If gdb crashes while you are trying to attach to a Java application, copy /System/Library/Frameworks/JavaVM.framework/Versions/A/Resources/MacOS/JavaApplicationStub into your application's bundle at <name> app/Contents/MacOS/JavaApplicationStub.

Java Events

Radar #4046940

DropTargetDragEvent.rejectDrag() did not have any visible effect

Description:

Calling rejectDrag() on a DropTargetDragEvent now properly rejects the drag, and drag object bounces back to its source.

Resolution:

This problem is now fixed.

Java Graphics

Radar #4488745

JViewport in BLIT_SCROLL_MODE don't paint properly with Sun2D renderer.

Description:

JViewport now paints correctly while scrolling with Sun2D renderer. The Sun2D renderer is now the default for Mac OS X v10.5.

Resolution:

This problem is now fixed.

Radar #4690482

ImageIO.read() does not properly decode TIFF images with Sun2D renderer.

Description:

ImageIO.read() now decodes TIFF images correctly with Sun2D renderer. The Sun2D renderer is now the default for Mac OS X v10.5.

Resolution:

This problem is now fixed.

Radar #5214320

Gradient fills of Java GeneralPath don't respect the even-odd winding rule for the Quartz pipeline.

Description:

Gradient fills of Java GeneralPath now works correctly with the even-odd winding rule for the Quartz pipeline. Quartz is not the default—the Sun2D renderer is now the default for Mac OS X v10.5.

Resolution:

This problem is now fixed.

Java HotSpot

Radar #3441352

Java crashes resulted in 2 separate crash logs: CrashReporter's log and the JavaNativeCrash log.

Description:

For Java 1.5 crashes, the CrashReporter crash log now adds all of the information that the JavaNativeCrash log previously did. There no longer is a JavaNativeCrash log.

Resolution:

This problem is now fixed.

Radar #4413936

Concurrent Mark Sweep Garbage Collection Issues

Description:

Applications that use the "+UseConcMarkSweepGC" flag (Concurrent Mark Sweep Garbage Collection algorithm) tended to be unstable.

Resolution:

This problem is now fixed.

Radar #4626636

On Intel Macs, using a custom JVMTI agent may cause crash.

Description:

Custom JVMTI agent native libraries will cause a crash due to EXC_BAD_INSTRUCTION if they call out to another library with a misaligned stack.

Workaround:

This crash can be avoided by adding the `__attribute__((force_align_arg_pointer))` gcc attribute to the declaration of each function in the custom JVMTI agent native library.

Radar #4896710

Java crashes when passed `-Dsun.boot.class.path` argument, if that path contains more than one directory

Description:

If a users passed `-Dsun.boot.class.path` with more than one directory on the path, Java would crash.

Resolution:

This problem is now fixed.

Java InputMethods

Radar #4693167

All but the first character is stripped from native key events

Description:

Keystrokes that generated multiple characters only generated a `KEY_TYPED` for the first character.

Resolution:

If a keystroke generates multiple characters (common on Hebrew and Cherokee Nation layouts) we now generate `KEY_TYPED` events for each character generated. If the keystroke went to a component that is accepting input method events, an `InputMethodEvent` of type `INPUT_METHOD_TEXT_CHANGED` with all of the characters is generated.

Radar #5158674

Incorrect key events generated for dead key strokes

Description:

Key events for the first part of a dead-key combination were not being generated correctly. For example, option + e was giving you `VK_DEAD_ACUTE + ALT`.

Resolution:

option + e now generates a `KEY_PRESSED` of `VK_E + ALT`, as expected. If no modifiers were pressed, and the key generates a dead key character we generate the dead key character as we did before.

Java JavaLib

Radar #4538717

Java encoding doesn't work as expected

Description:

The class name in Unicode should always be found and loaded, and be able to put into a jar file. It had been observed that when running programs in Java on Mac OS X some character sets didn't display properly and couldn't be found in the jar file.

Resolution:

This problem is fixed in Mac OS X v10.5

Radar #4963965

Java does not contain the latest time zone data

Description:

Canada changed its date for switching to Daylight Savings Time, as well as numerous other changes to time zone information.

Resolution:

Java 1.5 contains the most recent version of the Olson time zone data (2007g) from August, 2007.

Radar #5015600

Java 1.5 uses ShiftJIS encoding when the primary language is Japanese

Description:

By default, Java 1.5 would use ShiftJIS encoding when the primary language is set to Japanese. Now it uses UTF-8.

Resolution:

This problem is now fixed.

Radar #5461326

The KH characters are divided in Java Preferences

Description:

The KH characters is divided in Java Preferences.

Resolution:

This problem is now fixed.

Java JDK

Radar #3752549

Support .dylib as a native library suffix.

Description:

Prior to Mac OS X 10.5, the only extension for Java native libraries was jnilib.

Resolution:

The Java VM will first try to load a native library named lib<library>.jnilib. If that fails, it will look for a native library named lib<library>.dylib. If that fails, an UnsatisfiedLinkException is thrown.

Java PB Templates

Radar #4621312

Java-related Xcode projects need to be updated for compatibility and consistency

Description:

The Java Xcode templates used the legacy Java build system.

Resolution:

All new projects created with Xcode now use the Ant build system to build their Java components, which is now recommended for all Java development with Xcode.

Java Printing

Radar #4417162

Printing to shared printers did not work.

Description:

When attempting to print to a "Shared Printer" (instead of a "Bonjour Printer" or a manually added printer), the print job would be bounced to the default printer, or if no printers were set, it would be lost.

Resolution:

This issue has been resolved. Print jobs sent to printers shared through other computers now correctly print.

Radar #5275318

Shared CUPS printers with punctuation or non-roman characters were not accessible to Java.

Description:

Shared CUPS printers are now available to Java Printing, provided they are added in the Mac OS X "Printing & Fax" Preference Pane, regardless of any characters in the name.

Resolution:

This problem is now fixed.

Java Security

Radar #4562837

Root certificates can now have their own trust settings

Description:

In Mac OS X 10.5 users can now assign trust settings to root certificates in the System Roots keychain in Keychain Access.

Resolution:

The Java plugin relies on the Mac OS X security infrastructure to provide a list of trusted root certificates. If the user has changed any trust settings on the root certificates, they are honored when establishing trust for a signed JAR or https connection.

Java Sound

Radar #4170668

Java Sound Mixers for USB/FireWire Mics: request for improved device info.

Description:

Java Sound now provides a more descriptive info (device name and vendor name) for the Mixer object.

Resolution:

This problem is now fixed.

Radar #4985417

USB Microphones are not recognized as default input device.

Description:

Java Sound now returns the default input device as the first entry in an array specified by given selection criteria.

Resolution:

This problem is now fixed.

Java SWT Support

Radar #4458063

Can't use the AWT from a Carbon application

Description:

Prior to Mac OS X 10.5, native Carbon applications could not use JNI to access Java code that started the AWT.

Resolution:

Carbon applications can now access the AWT if they follow the guidelines in Tech Note 2147. Specifically, they must start the Java VM on a non-main thread, and must take care to only make Carbon or Cocoa calls on the main thread.

Radar #4997714

SWT applications that also create AWT windows hang on exit.

Description:

An SWT application that created an AWT window would not exit cleanly when the last window was closed. The spinning beach ball appeared and you had to force quit the application.

Resolution:

This problem is now fixed in Mac OS X 10.5. Quitting the application via the application menu or closing all windows now ends the application without any problems.

Java Text

Radar #5090338

Obtaining the GlyphVector outline of characters in certain fonts yielded "crumpled" outlines

Description:

Some PostScript fonts containing cubic paths were truncating some point information about the character stroke. This issue has been resolved for all line, quad, and cube path components.

Resolution:

This problem is now fixed.

Radar #5120375

Bi-directional character information did not render correctly in the Sun2D renderer

Description:

All bi-directional languages (Arabic, Hebrew, etc) should now render correctly in all Java text fields.

Resolution:

This problem is now fixed.

Java Toolkit

Radar #5186158

Can't disable Command-H and Command-Q as shown in TN2062

Description:

The techniques described in Section 13 of TN2062 no longer work with Java 1.4.2 or later. Attempting to get the Quit or Hide Application menu items fail with a return value of `menuItemNotFoundErr`.

Workaround:

Section 13 of TN2062 describes how to disable the Hide Application or Quit item for Carbon and Cocoa applications. Java 1.3.1's implementation of the AWT was based on Carbon, but Java 1.4.2 and later are based on Cocoa, so the Carbon examples in this section no longer apply. If you need to disable these menu items, create your own NIB file with Interface Builder, and set the property 'apple.awt.application.nib' to be the full path to the NIB file. You may wish to write a short native method that returns the path to the localized NIB file, and then set the value of that property to the path found.

Java Tools

Radar #3606823

The rights to redistribute the `JavaApplicationStub` launcher binary are unknown

Description:

The license to redistribute the `JavaApplicationStub` are available from the "JavaApplicationStub License" menu item in the "Help" menu of the Jar Bundler application in `/Developer/Tools`. The full text of the license is also available at `/System/Library/Frameworks/JavaVM.framework/Resources/Documentation/JavaApplicationStub-License.rtf`.

Resolution:

This problem is now fixed.

Radar #4511969

Java developers would like JUnit available on Mac OS X.

Description:

JUnit 4.1 is now included in the base install of Mac OS X 10.5 in `/usr/share/junit`.

Resolution:

This problem is now fixed.

Radar #4950984

Java developers would like Maven available on Mac OS X.

Description:

Maven 2.0.6 is now included in the base install of Mac OS X 10.5 in `/usr/share/maven`.

Resolution:

This problem is now fixed.

Java Web Start

Radar #4073782

Java Web Start could not be used to deploy Eclipse RCP applications that used SWT

Description:

Java Web Start was not able to correctly start up applications that used the SWT, even if you specified `-XstartOnFirstThread` as a VM argument.

Resolution:

Java Web Start now supports deployment of applications that use the SWT. See the instructions for using SWT in a JNLP-based application at <http://www.eclipse.org/swt/jws/>. You will also need to add a 'java-vm-args' attribute to the `j2se` element of the Mac OS X-specific resources block: `<resources os="Mac OS X"> <j2se version="1.5*" java-vm-args="-XstartOnFirstThread"/> </resources>`

Radar #4539542

JNLP applications won't run if download folder's name has a non-ASCII character in it

Description:

Java Web Start wasn't able to open a JNLP file if the folder it was downloaded to had a non-ASCII character in its name.

Resolution:

This problem is fixed in Mac OS X v10.5

Radar #4908673

Java Web Start crashed when opening some JNLP files

Description:

Java Web Start crashed if the `<description>` node of the `<information>` section was empty.

Workaround:

This issue has been resolved.

Document Revision History

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

Date	Notes
2007-10-31	First version of Java for Mac OS X v10.5 Release Notes.

REVISION HISTORY

Document Revision History