J2SE 5.0 Release 4 Release Notes

Java

2006-05-23

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

Simultaneously published in the United States and Canada.

Contents

Tables

Introduction to J2SE 5.0 Release 4 Release Notes

This release of Java for Mac OS X includes improvements for Java 2 Platform, Standard Edition 5.0 (J2SE 5.0) on Mac OS X. It features Apple's implementation of Sun's J2SE Version 1.5.0_06.

What is J2SE 5.0 Release 4 for Mac OS X?

J2SE 5.0 Release 4 for Mac OS X provides numerous enhancements and bug fixes for Apple's implementation of J2SE 5.0 on Mac OS X v.10.4. This release includes compatibility with Sun's Java 2 Platform Standard Edition, version 5.0 (1.5.0_06).

For general information about Java changes in J2SE 5.0, see *Release Notes - Java 2 SDK, Standard Edition Version 5.0* at http://java.sun.com/j2se/1.5.0/relnotes.html.

Note: After installing J2SE 5.0 Release 4, J2SE 5.0 becomes the preferred Java version, superseding Java 1.4.2. Java 1.4.2 is still installed on your Mac, but applications run with J2SE 5.0 unless they specifically request Java 1.4.2.

Who Should Read This Document?

Any developer who wants to distribute Java applications for Mac OS X should read this document since various issues and fixes found in this release may affect your application. Anyone interested in new Java development (either J2SE or Cocoa Java) should read this document for the most current information on new features and outstanding issues with Java on Mac OS X.

Organization of This Document

This document contains the following chapters:

- "Resolved Issues" (page 9) 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.
- "Outstanding Issues" (page 23) presents a selection of high-visibility bugs that you may need to work around with this release. This chapter is broken down by the category where the bug occurs and provides a brief description of what the issue is and often provides a workaround for the issue.

This document also contains a revision history.

If you are just beginning Java development for Mac OS X, you can probably just read the "Outstanding Issues" (page 23) chapter. Otherwise, it is recommended that Java developers read all chapters.

See Also

The Following Apple Java documentation may be helpful:

- *Java Development Guide for Mac OS X*
- *Java Property, VM Option, and Info.plist Key Reference for Mac OS X*
- Previous Java Release Notes
- *Java on Mac OS X Frequently Asked Questions* (http://developer.apple.com/java/faq/)

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 Applets

Radar #4398074

Java Preferences didn't save applet parameters

Description:

If the `deployment.properties` file didn't exist in `~/Library/Caches/Java/` prior to running Java Preferences, virtual machine parameters were not be saved until Java Preferences.app was used again.

Resolution:

The `deployment.properties` file is now properly created and saved.

Radar #4401040

All certificates treated as self-signed

Description:

The Java plug-in treated all certificates used with JAR files or HTTPS connections as self-signed certificates. This left the incorrect impression that the certificate was not issued by a trusted authority, advising the user to take additional caution.

Resolution:

The Java plug-in now correctly reads the system-wide trusted X.509 anchor certificates and displays certificates correctly.

Java Aqua Look and Feel

Radar #4400344

JFileChooser deadlock

Description:

Showing a `JFileChooser` from a non-event dispatch thread caused a deadlock.

Resolution:

When a `JFileChooser` is called from a non-event dispatch thread, the call is redirected to the proper thread.

Note: Calling a `JFileChooser` from a non-event dispatch thread is thread unsafe and is advised against by Sun guidelines. This circumstance is taken into account in J2SE Release 4 since it is very common. Despite this, it is recommended that you originate calls to a `JFileChooser` on an event dispatch thread.

Java AWT

Radar #4125987

Custom cursors causing exceptions

Description:

Attempting to use a custom cursor created using `Toolkit.createCustomCursor` with a `NULL` name triggered an exception.

Resolution:

Cursors with a `NULL` name work as expected.

Radar #4322814

AWT Choice control

Description:

When focus was on an AWT Choice control, pressing the space bar or down arrow key didn't show its popup menu.

Resolution:

An AWT Choice control now displays its popup menu when the space bar or down arrow key is pressed.

Radar #4323039

Focus between AWT and Swing controls

Description:

Tabbing between AWT and Swing controls in the same window caused focus to stick on some controls.

Resolution:

Focus shifts properly between peer AWT and Swing elements.

Radar #4326611

Applet dialog appearance

Description:

Dialogs appeared in Apple's textured style, regardless of the `apple.awt.UseBrushMetal` setting.

Resolution:

By default, dialogs appear using the Aqua Look and Feel.

Radar #4367914

Help menu issues

Description:

Modifying a Help menu with `setHelpMenu` would hide another menu from the Menu Bar.

Resolution:

Modifying help menus no longer causes other menus to disappear.

Java Events

Radar #4005998

Key binding conflicts

Description:

Key bindings for items in the Services menu weren't passed to Java applications after an item from the Application menu was chosen.

Resolution:

Key bindings for disabled items in the Services menu are always passed to Java applications. Bindings for enabled items may still conflict.

Java Graphics

Radar #4314252

Sun 2D Renderer support

Description:

Previous releases of Java on Mac OS X only offered 2D rendering support using Apple's Quartz technology.

Resolution:

J2SE 5.0 Release 4 features support for Sun's 2D renderer. By default, the Quartz renderer is used for 2D rendering. You can use the Sun 2D renderer by setting the `apple.awt.graphics.UseQuartz` runtime option to `false`.

Radar #4323478

Graphics corruption during live resize

Description:

Overriding the `update()` method and not calling `super.update()` in a heavyweight component may have resulted in a blank window during a live resize.

Resolution:

All components redraw correctly during a live resize.

Radar #4347718

Window disposal crash

Description:

Disposing a window while it's being drawn to caused a crash.

Resolution:

An application no longer crashes if it disposes a window while it's being drawn to.

Java Libraries

Radar #4399262

Applications that depend on org.apache packages

Description:

Sun moved the `org.apache` package in J2SE 5.0. Various applications depend on the org.apache package and therefore break when run under J2SE 5.0.

Resolution:

J2SE 5.0 Release 4 includes the `org.apache` package to provide increased backward compatibility. The compatibility classes are appended to the end of the `CLASSPATH` so newer versions of the `org.apache` XML classes can be used.

Note: Applications should not depend on this functionality's presence in future releases of Java on Mac OS X.

Radar #4404074

Runtime.exec and accented characters

Description:

If the name of the executable or its path had accented characters, `Runtime.exec` failed to execute it.

Resolution:

`Runtime.exec` handles paths and executable names that contain accented characters.

Java Printing

Radar #2937917

Page orientation not selectable

Description:

There was no way to select the page orientation when using `Toolkit.defaultToolkit().getPrintJob()`.

Resolution:

The Page Setup dialog appears after selecting Print from the Print dialog.

Radar #4367998

Printing attributes ignored

Description:

When using `JTable.print()` or the `javax.print` dialog, set job attributes were not passed to `PrinterJob`.

Resolution:

Job attributes (like copies and orientation) are now properly set.

Radar #4401516

Wrong dialog when printing to file

Description:

When selecting Print to File from the `javax.print` dialog, an Open file dialog was shown.

Resolution:

Selecting Print to File displays a Save dialog.

Radar #4417162

Shared printers unsupported

Description:

Attempting to print to a shared printer redirected the print job to the default printer.

Resolution:

Print jobs sent to printers shared through other computers print on the shared printer.

Java Security

Radar #3173179

Restrictions on cryptographic operations

Description:

It was necessary to install Sun's unlimited strength jurisdiction policy files if there was a need for cryptographically strong certificates or algorithms.

Resolution:

Sun's unlimited strength jurisdiction policy files are now pre-installed.

Radar #4115657

KeychainStore and private keys

Description:

`KeychainStore` could not read or write the `Key` portion of private keys stored in the user's keychain.

Resolution:

`KeychainStore` can read private keys and certificate chains that make up an identity from Keychain. Also, PKCS#12-formatted identities can be stored and retrieved from Keychain.

Radar #4185385

Certificate trust alert clarification

Description:

The certificate trust alert was unclear that running the signed code granted the issuing application or applet unlimited access to the user's machine.

Resolution:

The certificate trust alert has been reworded to emphasize the consequences of clicking Trust. If your documentation used a screen shot of this alert, you should update it.

Radar #4280141

Certificate trust alert ownership

Description:

When a signed applet is loaded, the certificate trust alert is presented as a modal dialog. If the window with the signed applet was in the background, the dialog was shown but no indication was made to associate the alert and its applet.

Resolution:

The applet's window is brought to the foreground before presenting the certificate trust dialog. Also, the alert's title is set to the URL of the page hosting the applet.

Radar #4404315

PKCS#11 support missing

Description:

J2SE 5.0 supports native PKCS#11-based smartcard libraries. The required Jar and native library were not included with previous releases of J2SE 5.0 for Mac OS X.

Resolution:

J2SE 5.0 Release 4 includes PKCS#11 support.

Radar #4422901

KeychainStore loses certificates

Description:

Multiple operations on `KeychainStore` left it in an inconsistent state where certificates added by the developer may have been lost.

Resolution:

J2SE 5.0 release 4 includes a more robust implementation of `KeychainStore`.

Radar #4078337

Certificates for SSL authentication

Description:

Previous versions of J2SE 5.0 on Mac OS X didn't look in Keychain for certificates or keys for sites that used SSL authentication.

Resolution:

In J2SE 5.0 Release 4, the `KeychainStore` implementation is used to find certificates and keys requested by a server for SSL authentication. You can also continue to use a Java keystore (found at `~/Library/Caches/Java/security/trusted.clientcerts`), but Keychain is the preferred location to store certificates and keys.

Java Swing

Radar #4404084

JPasswordField with Input methods

Description:

Input using Input methods while a `JPasswordField` had focus would send entered characters to the previously focused component.

Resolution:

Text entered into a `JPasswordField` properly dispatches each key-press into the `JPasswordField`, regardless of input method. Also, AWT correctly respects the use of `enableInputMethods(false)` on a Component.

Java SWT Support

Radar #3905894

AWT integration within SWT applications

Description:

SWT applications were unable to host AWT or Swing windows.

Resolution:

J2SE 5.0 allows SWT applications to host AWT and Swing windows. Java Web Start applications and Applets are unsupported.

This fix is provided by using the Compatibility Mode feature of `CocoaComponent` and is subject to many of its limitations. Also, you cannot embed AWT and Swing components within SWT windows. Finally, there are known issues with modal dialogs. To reliably show dialogs, use `SwingUtilities.invokeLater`.

Java Text

Radar #4298266

Type 1 fonts not loaded

Description:

Type 1 font files (pfa and pfb) were not loaded by `FontManager`.

Resolution:

The Sun 2D renderer, as discussed in “Sun 2D Renderer support,” supports Postscript Type 1 fonts.

Using the Quartz renderer allows you access the character to glyph mappings and some measurement data, but attempting to draw with a Type 1 font results in unexpected glyphs drawn in another font.

Radar #4298327

Custom font paths ignored

Description:

Paths specified using the `sun.java2d.fontpath` runtime option were not scanned for additional fonts to add to a Java application's font list.

Resolution:

Fonts in directories specified using the `sun.java2d.fontpath` runtime option are available.

Radar #4337704

Text in Metal-themed JTextAreas

Description:

When selecting text from the end of a line forward, the characters in the text moved and "shimmered" as the selection size changed.

Resolution:

The measurements for aliased text now align to the bounds of the displayed characters.

Radar #4367820

Font.canDisplay returns true

Description:

`Font.canDisplay` always returned true, even if the `Font` didn't have the requested glyph and Mac OS X character substitution couldn't find an appropriate substitute.

Resolution:

`Font.canDisplay` accurately returns whether or not a glyph can be displayed using a particular font. Note that there is no guarantee that the glyph comes from the requested font, only that a character can be displayed.

Radar #4371847

AWT controls use incorrect font

Description:

The order of font loading affected the font displayed on AWT controls, resulting in controls using Lucida Sans instead of Lucida Grande under some circumstances.

Resolution:

All AWT controls use Lucida Grande.

Radar #4390589

GlyphVector and AffineTransforms

Description:

Moving an individual glyph in a `GlyphVector` using the translation component of an `AffineTransform` didn't result in a translated glyph when the outline of the glyph was requested.

Resolution:

Translation components of `AffineTransforms` are applied to text drawn with `GlyphVector.getOutline`, `Graphics.drawGlyphVector`, and measurements of the `GlyphVector`.

Radar #4401414

Font style in name

Description:

Asking for an italic variant of a font by appending *-Italic* to it's name didn't provide an italic version.

Resolution:

Italic and bold italic variants are now returned by appending a space or a dash and the requested style.

Radar #4402021

Fractional font sizes

Description:

When drawing with a fractional size font using `Font.deriveFont`, measurements using the font would reflect the fractional size, however the actual drawing would only draw at the nearest integer size.

Resolution:

This problem has been fixed for all text drawing operations, such as `Graphics.drawString`, `Graphics.drawGlyphVector`, etc.

Java Virtual Machine

Radar #4383089

Incorrect code generation for arraylength bytecode

Description:

Code generated for the `arraylength` bytecode caused an `IndexOutOfBoundsException` to be thrown when using various `Zip` APIs.

Resolution:

The correct native code is generated.

Java Web Start

Radar #3655021

Splash icons used as Dock icon

Description:

If the first `icon` element in an `information` descriptor was of type `splash`, that element was used as the application icon, even if an icon of the correct type (`default` or none) was also listed.

Resolution:

The first icon with no type or of type `default` is used as the application icon.

Radar #4075884

JNLP settings not saved

Description:

If a JNLP file specified a custom memory setting or a property that would normally be set from the command line, the setting was not copied into the `Java` dictionary of the `Info.plist` when a desktop application was created.

Resolution:

When creating a desktop application, custom memory settings, safe VM options, and safe properties from the JNLP file are copied into the application's `Info.plist`.

Radar #4086608

Java Web Start man page

Description:

A number of the paths referred to in the Java Web Start man page were incorrect.

Resolution:

The Java Web Start man page has been updated.

Other Resolved Issues

Table 1-1 (page 19) lists numerous issues present in previous versions of Java for Mac OS X that are resolved in J2SE 5.0 Release 4.

Table 1-1 Various Resolved Issues in J2SE Release 4

Radar #	Description
4334236	Printing from an applet caused an exception

Radar #	Description
4385058	Java plug-in exception caused hang in Web Kit
4314206	Custom Nib files that caused exceptions left unloaded
3994051	Menu bar menus didn't flash when activated by keyboard shortcuts
3196013	Overlapping Containers' clipping not maintained
4336070	QuickTime for Java used `com.apple.eawt.CocoaComponent.CompatibilityMode=true`
4357922	Corrupted graphics after a live resize
4364454	`com.apple.eawt.CocoaComponent.CompatibilityMode` dismissed modal dialogs prematurely
4367748	Keyboard focus lost if focused component was hidden
4432247	Modal dialogs not always on top of other windows
4434041	`JNI_GetCreatedJavaVMs` returned a non-zero value under Rosetta
3589153	Focus ring not properly adjusted when AWT components were resized or moved
4306310	Shift-delete didn't delete characters in a `JTextComponent`
4388549	Flashing when live-resizing `JFrame` with overridden `paint` method
4372370	Passing in `-XX:+AggressiveHeap` flag caused crash
4383545	VM exception handler crash when converting a fault into a Java exception
4418700	Prebinding warning always shown
4329506	Unable to move cursor within a text field before hitting enter when using Input methods
4357153	Java Developer package removed previously installed Javadoc's `index.html`
4208349	`Locale.defaultLocale` incorrect
4408322	`Runtime.exec` didn't handle accented environment variables
4350107	`JNI_CreateJavaVM` used Java 1.4.2 despite system-wide default JVM setting of J2SE 5.0
4336058	Modal dialogs disappear in applications that use JOGL
4307013	Keyboard focus lost when switching between AWT and Swing components
4421110	False error message when using a self-signed certificate
4365514	Eclipse-based applications that don't use SWT wouldn't launch
4451025	AWT renamed threads it doesn't own
3997910	Transparency not respected when a GIF or PNG image used as an application icon

Radar #	Description
4263130	Messages to `System.out` or `System.err` in an unsigned Java Web Start application triggered a security exception
4345426	Safe properties in a Java Web Start application terminated the application with an `UnsatisfiedLinkException`
4450344	Calling `JViewport.setViewPosition` resulted in flickering and un-smooth painting
4114007	`JSeparator.setVisible` had no effect when the screen Menu Bar was enabled
4191032	Help tags assigned to a `JSpinner` were not shown

Outstanding Issues

This chapter provides a listing of bugs that you may need to work around in your Java code for Mac OS X. Where possible, workarounds are provided.

Java Application Support

Radar #4443822

EOModeler incompatible with J2SE 5.0

Description:

Older versions of the EOModeler application, bundled with WebObjects, are incompatible with J2SE 5.0 Release 4.

Workaround:

EOModeler needs to be forced into using Java 1.4.2. To do this, open its `Info.plist` file (located in `EOModeler.app/Contents/`) and change its `JVMVersion` key to `1.4*`.

Radar #4413220

Borland JBuilder 2005 incompatible with J2SE 5.0 Release 4

Description:

JBuilder 2005 update 4 added Borland's own implementation of the AWT classes based on Java 1.4.2. Concurrently, JBuilder uses a custom launcher that loads the VM using the `CurrentJDK` link. Since the VM is running J2SE 5.0 but the actual AWT classes are from Java 1.4.2, an exception is thrown.

Workaround:

This issue is unresolved at this time. Reverting to an older version of JBuilder 2005 is known to work.

Radar #4430398

OpenBase JDBC and J2SE 5.0

Description:

Database transactions with OpenBase fail with strange data mapping errors.

Resolution:

Download the latest J2SE 5.0-compliant OpenBase JDBC driver at http://www.openbase.com.

Java AWT

Radar #4115400

CFMessagePortCreateLocal error message

Description:

When running a second Java process from the command line, the error message `CFMessagePortCreateLocal(): failed to name Mach port (java.ServiceProvider)` may be shown.

Workaround:

This error is harmless and can be ignored.

Java Developer

Radar #4369097

Xcode Java templates don't build and run successfully

Description:

Java templates in Xcode compile using the current JDK but use a `JVMVersion` key of `1.4*`. This means that after you install J2SE 5.0 Release 4, projects compile with J2SE 5.0 and try to launch using Java 1.4.2.

Workaround:

To work around this issue:

- *To compile and run with Java 1.4.2:* Under Java Compiler Settings for the build target, Set *Target VM version* and *Source Version* to `1.4*`
- *To compile and run with J2SE 5.0:* Under Pure Java Specific in the project settings, set *Target VM Version* to `1.5*` or `1.5+`.

Radar #4397768

Java developer examples compiled for J2SE 5.0

Description:

The examples in `/Developer/Examples/Java/` are built with the default compiler, which is now J2SE 5.0. If you compile with the default settings, your build results can only be run on machines with J2SE 5.0 installed.

Workaround:

To compile a sample for use on an earlier version of Java, set the "Target VM version" and "Source Version", available under Java Compiler Settings for the build Target, to `1.4*`.

Java Libraries

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.

Java Virtual Machine

Radar #4413936

Non-default Garbage Collection issues

Description:

Applications that use the `-XX:+UseConcMarkSweepGC` (Concurrent Mark Sweep Garbage Collection algorithm) or `-XX:+useParallelGC` (Parallel Garbage Collection algorithm) tend to be unstable.

Workaround:

None.

Radar #4497576

Java troubleshooting tools non-functional

Description:

Java troubleshooting tools, like `jinfo`, `jmap`, and `jstack`, don't work when run as a regular user.

Workaround:

To use these tools, prepend them using the `sudo` command. You need to kill the `hsdbd` process before using any of these commands with sudo.

Document Revision History

This table describes the changes to *J2SE 5.0 Release 4 Release Notes*.

Date	Notes
2006-05-23	Radar #4307013 was incorrectly included as an Outstanding Issue.
2006-04-21	First draft of J2SE 5.0 Release 4 Release Notes

Document Revision History