

---

# Java 1.4.2 Release Notes

Java



2004-08-11



Apple Inc.  
© 2004 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, Cocoa, ColorSync, Mac, Mac OS, Panther, Safari, WebObjects, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

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

Times is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

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 1.4.2 Release Notes** 7

---

Who Should Read This Document? 7  
Organization of This Document 7  
See Also 8

## **Chapter 1**      **What's New in Java 1.4.2** 9

---

How and Where to Get Java 1.4.2 9  
Java 1.4.2 Replaces Java 1.4.1 9  
New LiveConnect Support 10  
New JFileChooser Dialogs 10  
New ColorSync CMMs 11  
AWT Enhancements 12  
Proxy Server Enhancements 12

## **Chapter 2**      **Resolved Issues** 13

---

Java Applets 13  
Java Application Support 13  
Java Aqua LAF 14  
Java Events 14  
Java HotSpot 15  
Java JavaLib 16  
Java Tools 16  
Java Web Start 16

## **Chapter 3**      **Outstanding Issues** 17

---

Java Application Support 17  
Java AWT 17  
Java Graphics 18  
Java HotSpot 19  
Java Security 19  
Java Swing 20  
Java Text 21

## **Document Revision History** 23

---



# Figures and Tables

## Chapter 1

## What's New in Java 1.4.2 9

---

Figure 1-1	Before: Java 1.4.1 open dialog	11
Figure 1-2	After: Java 1.4.2 open dialog	11
Table 1-1	Values to specify the Java version in Mac OS X	9



# Introduction to Java 1.4.2 Release Notes

---

This release of Java for Mac OS X brings compatibility with Sun's Java 2 Platform, Standard Edition Version 1.4.2.

## What is Java 1.4.2?

Java 1.4.2 in Mac OS X is Apple's release of a Java VM and class libraries that are compatible with Sun's Java 2 Platform Standard Edition, version 1.4.2. Sun's Java 1.4.2 release has included many changes that have also been incorporated in the Java 1.4.2 release of Java for Mac OS X. In addition, it adds a few Mac OS X-specific enhancements and addresses some bugs present in the Java 1.4.1 release for Mac OS X. For general information about Java changes in Java 1.4.2, see *Enhancements and Changes in Java 2 SDK v1.4.2* at <http://java.sun.com/j2se/1.4.2/changes.html>.

## Who Should Read This Document?

Any developer that currently distributes Java 1.4.1 applications for Mac OS X should read this document since the changes to Java 1.4.2 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 in Mac OS X.

## Organization of This Document

This document contains the following chapters:

- [“What's New in Java 1.4.2”](#) (page 9) discusses high level changes from the Java 1.4.1 release for Mac OS X to the Java 1.4.2 release.
- [“Resolved Issues”](#) (page 13) 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 17) 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.

If you are just beginning Java development for Mac OS X you can probably just read the [“Outstanding Issues”](#) (page 17) chapter. Otherwise it is recommended that Java developers read all three chapters.

## See Also

The Following Apple Java documentation may be helpful:

- *Java Development Guide for Mac OS X*
- *Java 1.4 System Properties*
- *Java 1.4 Virtual Machine Options*
- *Java on Mac OS X Frequently Asked Questions* (<http://developer.apple.com/java/faq/>)

The following Sun documents are especially helpful in determining how new features introduced in Java 1.4.2 may affect your applications:

- *Enhancements and Changes in Java 2 SDK v1.4.2*, available at <http://java.sun.com/j2se/1.4.2/changes.html>
- *Release Notes Java 2 SDK, Standard Edition Version 1.4.2*, available at <http://java.sun.com/j2se/1.4.2/rel-notes.html>



# What's New in Java 1.4.2

This chapter discusses some high level and highly-visible changes from the Java 1.4.1 release of Java for Mac OS X to the Java 1.4.2 release.

## How and Where to Get Java 1.4.2

Java 1.4.2 is available only for Mac OS X version 10.3 (Panther). It is not available for earlier versions of Mac OS X. There are two parts to the Java 1.4.2 release for Mac OS X, the user and developer packages. The user package is available through Mac OS X's Software Update and online at <http://www.apple.com/downloads/mac-osx/>. People that use Software Update will see this as a recommended update.

Apple does not provide a redistribution license for Java 1.4.2. Your customers need to download it directly from Apple's site.

The developer package is an optional update available from <http://connect.apple.com>. It includes updated developer examples, tools, project templates, and Javadoc API references for both Sun's and Apple's classes. Although optional, this is a recommended update for anyone doing Java development for Mac OS X.

## Java 1.4.2 Replaces Java 1.4.1

It is important to recognize that with the installation of Java 1.4.2, Java 1.4.1 is no longer available. Java 1.4.2 replaces Java 1.4.1. If you have hard-coded any dependencies to Java 1.4.1, make sure to test your code on Mac OS X with Java 1.4.2.

If you have wrapped your Java application as a native Mac OS X application bundle, verify that you are using the proper Java version string in your Info.plist file. Because Mac OS X still includes Java 1.3.1 it is important that you specify the correct version of Java for your application to run in. Table 1-1 shows the valid values for the Java version key and indicates the result of using that value.

**Table 1-1** Values to specify the Java version in Mac OS X

String	Java version used	Notes
1.3.1	1.3.1	Specifies an exact version of Java. It is not recommended that you use this key unless absolutely necessary.
1.3*	1.3.1	Requests the highest version of Java 1.3 available. Note that if Java 1.3 is updated in future releases of Mac OS X, the latest version of Java 1.3 will be used.
1.3+	1.4.2	Requests the highest version of Java above 1.3. Note that if Java is updated in future releases of Mac OS X, the latest version of Java will be used.

String	Java version used	Notes
1.4.1	1.4.2	With Java 1.4.2 is installed, Java 1.4.1 is no longer available. For compatibility, applications that specify 1.4.1 will run in Java 1.4.2. You should update these applications to specify 1.4* or 1.4+ as appropriate since the designation of a specific version has been deprecated.
1.4*	1.4.2	Specifies the highest version of the Java 1.4 available. Note that if Java 1.4 is updated in future releases of Mac OS X, the latest version of Java 1.4 will be used.
1.4+	1.4.2	Specifies the highest version of Java above 1.4. Note that if Java is updated in future releases of Mac OS X, the latest version of Java will be used.

**Important:** Don't specify a specific version of the VM to use unless absolutely necessary. It is much safer to use either a star (\*) or a plus (+) sign to indicate a version family rather than a specific version.

## New LiveConnect Support

Java 1.4.2 brings support for a subset of LiveConnect functionality to Safari. Safari, version 1.2 or greater, can now allow JavaScript on a web page to interact with Java applets and for Java applets to interact with JavaScript. Other features associated with LiveConnect, such as the ability for JavaScript to interact with arbitrary Java classes, are not supported in this release.

## New JFileChooser Dialogs

JFileChooser dialogs in Mac OS X have never looked very much like their native counterparts. With the Java 1.4.2 release, JFileChoosers have been updated so that, while they still retain full Java functionality, they now integrate more cleanly with native file chooser dialogs. Figure 1-1 and Figure 1-2 illustrate the old and new dialogs respectively. Keep in mind that the AWT file chooser provides the best native experience in Mac OS X. Unless you absolutely need to use the Swing file chooser, try to use the AWT file chooser (FileDialog).

Figure 1-1 Before: Java 1.4.1 open dialog

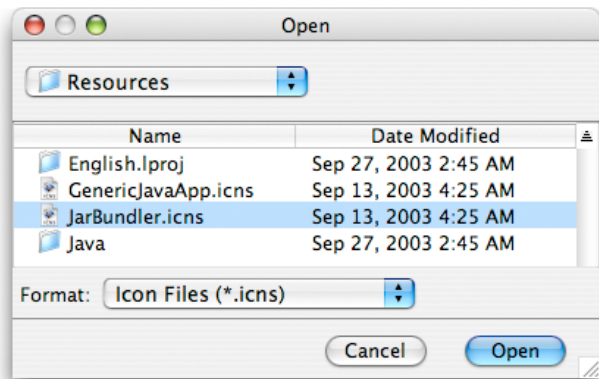
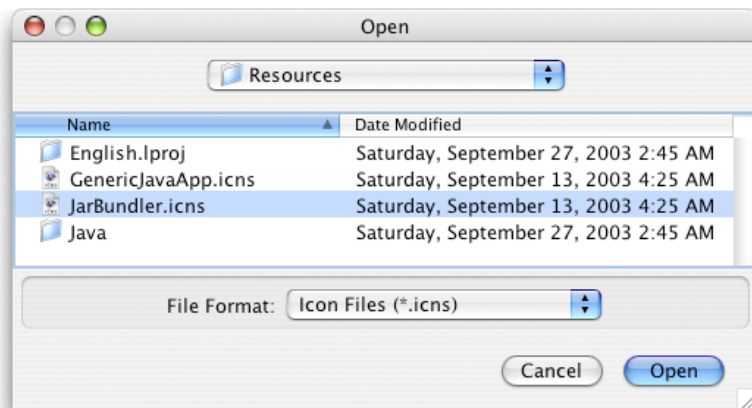


Figure 1-2 After: Java 1.4.2 open dialog



## New ColorSync CMMs

In Java 1.4.1 for Mac OS X, the `java.awt.color` native code was implemented using the KodakCMM codebase. While this is still the default `ICC_ColorSpace` implementation, with Java 1.4.2 you may also take advantage of a ColorSync implementation. Using ColorSync is especially useful if you want to take advantage of ColorSync profiles Mac OS X users may already be using in their existing workflow. If you are more concerned about cross platform compatibility, you should probably use the default Kodak CMM. Enable the ColorSync implementation with the `apple.cmm.usecolorsync` system property. For example, to enable it from the command line you could pass the following flag to `java`:

```
-Dapple.cmm.usecolorsync=true
```

## AWT Enhancements

Many AWT enhancements have been included with this release. In general, your users should see increased AWT robustness and faster AWT startup times.

## Proxy Server Enhancements

Several issues were raised with Java proxy server support with Java 1.4.1 in Mac OS X. The Java 1.4.2 has incorporated the following specific improvements:

- FTP and SOCKS connections now properly respect the systems proxy exception list.
- The operating system's HTTPS proxy server is now used correctly by Java applications.
- Java applications now use the operating system's proxy exception list like native applications.

In general, proxy support is much more integrated with the operating system's proxy support and Java applications should respect the settings in the Proxies pane of Network Preferences.

# 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 #3438550**

---

Safari sometimes crashes when users click the Back button to return to a page that contains an applet.

**Description:**

If a page with an applet is loaded, another page is loaded, and then the Back button is used to return to the page with the applet, Safari may crash.

**Resolution:**

This has been corrected in Java 1.4.2 and Safari should no longer crash.

## Java Application Support

### **Radar #3530571**

---

JBoss and Tomcat do not start after installing the Java 1.4.2 update.

**Description:**

The run.conf script for starting the JBoss server has a hard-coded reference to the 1.4.1 JVM. Installing the Java 1.4.2 update removes the Java 1.4.1 installation and JBoss/Tomcat services will not be able to launch. This is applicable only for Mac OS X users who have installed the Java Application Servers Development package from the Xcode Tools or users of Mac OS X Server.

**Resolution:**

For Mac OS X Server, download the JBoss update available through Software Update or from <http://www.info.apple.com/support/downloads.html>. For Mac OS X, open /Library/JBoss/3.2/bin/run.conf in a text editor and change the line:

```
JAVA=/System/Library/Frameworks/JavaVM.framework/Versions/1.4.1/Home/bin/java
```

to

```
JAVA=/usr/bin/java
```

**Radar #3549675**

---

WebObjects 5.2 is not supported with Java 1.4.2

**Description:**

Although there are no known issues with WebObjects 5.2 and Java 1.4.2, WebObjects 5.2 has not been qualified for Java 1.4.2. This means that you may experience unexpected and unsupported issues with WebObjects 5.2 and Java 1.4.2. For example, Java 1.4.2 has a different XML parser than 1.4.1 and could cause problems serializing/deserializing WebObjects classes.

**Resolution:**

Qualification is being evaluated for a future update or release of WebObjects.

## Java Aqua LAF

**Radar #3335189**

---

ComboBox.font and TableHeader.font do not return correct fonts.

**Description:**

In the Aqua LAF, the UIDefaults ComboBox.font is 14-point Lucida Grande Regular. The Apple Human Interface Guidelines specify that this value should be 13-point Lucida Grande Regular. The UIDefaults TableHeader.font is 13-point Lucida Grande Regular. The Apple Human Interface Guidelines specify that this value should be 11-point Lucida Grande Regular.

**Resolution:**

This has been corrected. ComboBox.font now gives 13-point Lucida Grande Regular and TableHeader.font now gives 11-point Lucida Grande Regular.

**Radar #3406630**

---

NullPointerException when double-click on an alias.

**Description:**

In JFileChooser, if a user double clicks on an alias, Java would throw a NullPointerException.

**Resolution:**

This has been fixed in Java 1.4.2; users can now safely double-click aliases in JFileChooser dialogs.

## Java Events

**Radar #3164718**

---

control-drag generates mouseMoved, not mouseDragged.

**Description:**

Moving the cursor with the Control key and the mouse key held down generates mouseMoved events.

**Resolution:**

Do not use the Control key as the modifier. On Mac OS X, the Control key in conjunction with a mouse click opens a contextual menu.

### Radar #3185427

---

KeyEvent.consume does not work in TextFields and TextAreas.

**Description:**

In java.awt.TextFields or java.awt.TextAreas (or classes derived from them), when consume is called on the KeyEvent in processKeyEvent it does not function properly. It should prevent the character being added to the text component, but does not and the character is added.

**Resolution:**

This has been fixed in Java 1.4.2. The correct behavior is now observed where characters are no longer added when consume is called.

### Radar #3222932

---

Java does not generate keyLocation values.

**Description:**

In previous releases the ability to distinguish between the left and right Shift, Control or Alt (Option) keys was not supported.

**Resolution:**

The left and right locations are now reported correctly on all keyboards (USB and ADB) where the hardware can tell the difference. Some keyboards cannot tell the difference and always return KEY\_LOCATION\_LEFT. Note that you should *not* bind any action to a specific (left or right) version of these modifier keys. It assumes key layout that may not be present on all keyboards.

## Java HotSpot

### Radar #2951807

---

The MaxFDLimit flag is not active in Mac OS X

**Description:**

Java on Solaris includes a flag (MaxFDLimit) to increase the file descriptor limit to the maximum. Mac OS X should include this flag as well.

**Resolution:**

This option is now available for the HotSpot VM in Java 1.4.2 as the `-XX:+ MaxFDLimit` flag to the java command. See *Java 1.4 VM Options* for more information on this and other options available to the Java VM.

## Java JavaLib

### Radar #3262614

---

Java should use `kqueue()` instead of `select()`.

**Description:**

Mac OS X allows the use of the `kqueue()` call. Since this allows for faster performance of Networking and NIO, it should be used instead of `select()`.

**Resolution:**

In Java 1.4.2 `kqueue()` is used by default. If you want your applications to use `select()` instead, set the `java.nio.preferSelect` system property to `true`.

## Java Tools

### Radar #3437509

---

`gdb` is the default Java debugger in Xcode AWT projects .

**Description:**

If you create a new Java AWT Applet project in Xcode, the default debugger is `gdb`. Therefore, breakpoints in Java code are not observed.

**Resolution:**

In the "Groups & Files" pane of the project window there should be a group of executables. Open this group and double-click on the `appletviewer` executable. A window opens. Near the bottom of that window is a group of Launch Configuration settings. Change the "Launch using" setting to "Java Debugger."

## Java Web Start

### Radar #3185452

---

Security Advisory dialog box hangs application

**Description:**

In the Java Web Start Application Manager, when Security Advisory dialogs were dismissed, it could cause the application to freeze. The only way to recover at this point was to force quit the application.

**Resolution:**

This has been resolved in Java 1.4.2.



# Outstanding Issues

---

This chapter provides a listing of bugs that you may need to work around in your Java 1.4.2 code for Mac OS X. Where possible, workarounds are provided.

## Java Application Support

### **Radar #3522195**

---

Older versions of JBuilder may not work with Java 1.4.2.

**Description:**

Versions of JBuilder earlier than JBuilder X may not launch with Java 1.4.2 installed.

**Workaround:**

There is no known workaround other than upgrading to JBuilder X. You might also want to contact Borland technical support.

### **Radar #3524507**

---

Some Java applications that ran in Java 1.4.1 no longer run in Java 1.4.2.

**Description:**

If Java3D or JavaAI has been installed after installing Java 1.4.2 was installed, it's possible that certain applications double-clickable Java applications will not launch. The console may display `java.lang.UnsupportedClassVersionError`. Applications that will not run are those that explicitly specify Java 1.4.1 in the `Info.plist` file.

**Workaround:**

A user can simply reinstall Java 1.4.2. As a developer, you should make sure that the `info.plist` entry for `JavaVM` does not explicitly specify `Java1.4.1`. If so, change the setting to something safer like `Java1.4.1+` or `Java1.4*`.

## Java AWT

### **Radar #3114971**

---

Memory leak when creating and disposing `java.awt.Windows`

**Description:**

If you create new window objects then dispose of them, over time memory allocations increase.

**Workaround:**

There is currently none.

## Java Graphics

**Radar #3160445**

---

Apparent random crashes in heavily multi-threaded Java applications.

**Description:**

Some multi-threaded applications that use Java2D can crash with a crash log that specifies a failure in `Java_apple_awt_CRenderer_doFillRect`.

**Workaround:**

Currently there is none.

**Radar #3163400**

---

Rare deadlocks in multithreaded draw.

**Description:**

In certain situations, multithreaded applications that hold onto a graphics object before passing it to a different thread and then try to draw back into a AWT component will deadlock.

**Workaround:**

InvokeLater with the windows graphics object, which puts you on the AWT thread.

**Radar #3295811**

---

Problems with full screen mode.

**Description:**

Full screen mode has various problems in this release. For example, sometimes it will not initialize and when initialized it does not accept mouse events.

**Workaround:**

There are no, known workarounds.

**Radar #3296635**

---

Problems with XOR mode.

**Description:**

Using `setXORMode` does not always work correctly and you may see artifacts along the edges of shapes.

**Workaround:**

There are still outstanding issues with XOR mode in Java 1.4.2, but many issues can be resolved by using `setComposite(AlphaComposite.Xor)` where you might normally use `setXORMode(color)`.

## Java HotSpot

**Radar #3262798**

---

Problems using `-Xrunhprof` in Java Plug-in Control Panel.

**Description:**

Trying to use `-Xrunhprof` within the Advanced > Java Runtime Parameters field of Java 1.4.1 Plugin Settings results in only minimal usable data. A small amount of usable data is shown, then repeated messages of, "thread\_suspend failed."

**Workaround:**

There is no complete workaround, but `-Xprof` might give some useful results.

**Radar #3499564**

---

`jvmstat` tools do not work.

**Description:**

The `jvmstat` 2.0 tools available from <http://developers.sun.com/dev/coolstuff/jvmstat> do not work with Java 1.4.2 in Mac OS X.

**Workaround:**

None.

## Java Security

**Radar #3173133**

---

Java Kerberos does not work well with the default Mac OS X Kerberos implementation.

**Description:**

There are known issues with the interaction of Java and the operating system when dealing with Kerberos authentication. The Java environment is unable to correctly locate the credentials cache or tickets on the system.

**Workaround:**

None.

## Java Swing

### Radar #2884768

---

JMenu.getLocationOnScreen reports incorrect value for screen menu bar.

**Description:**

When the menu bar is set to be at the top of the screen `apple.laf.useScreenMenuBar`, `getLocationOnScreen` returns the location of JMenu components as if they were not in the menu bar.

**Workaround:**

None.

### Radar #3517451

---

Font style changes to Italic and Italic & Bold do not work for Dialog and Serif mode.

**Description:**

Microsoft Office installs fonts into the `Library/fonts/` folder in the user's home directory. Once installed, these fonts are used by Java applications instead of those installed with the system which may cause problems. The installed fonts include:

- Wingdings
- Verdana
- Times New Roman
- Tahoma
- Monotype Sorts
- Lucida Handwriting
- Copperplate Gothic Light
- Copperplate Gothic Bold
- Impact
- Edwardian Script ITC
- Curlz MT
- Century Gothic
- Arial
- Comic Sans MS
- Arial Black

**Workaround:**

Remove the fonts installed by Microsoft Office that are causing problems.

## Java Text

### **Radar #2826318**

---

Double-byte characters are not displayed correctly during input.

**Description:**

Double-byte characters don't display correctly when typed; instead they are shown as rectangles. This happens when the default Roman script system is the active script. This only happens while typing, once the character is committed, it appears correctly.

**Workaround:**

None.



# Document Revision History

---

This table describes the changes to *Java 1.4.2 Release Notes*.

Date	Notes
2004-08-09	Fixed links to other documents.
2004-02-04	Added information about WebObjeccts compatibility.
2004-02-02	First version of <i>Java 1.4.2 Release Notes</i> .

**REVISION HISTORY**

Document Revision History