Java 1.4.2 Update 1 Release Notes

Java

2004-08-11

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

Simultaneously published in the United States and Canada.

Contents

Tables

Introduction to Java 1.4.2 Update 1 Release Notes

This release of Java for Mac OS X brings compatibility with Sun's Java 2 Platform, Standard Edition Version 1.4.2_05.

What is Java 1.4.2 Update 1?

Java 1.4.2 Update 1 for Mac OS X is Apple's release of a Java virtual machine and class libraries that are compatible with Sun's Java 2 Platform Standard Edition, version 1.4.2_05. Sun's Java 1.4.2_05 release has included many changes that have also been incorporated in the Java 1.4.2 Update 1 release of Java for Mac OS X. In addition, it adds a few Mac OS X-specific enhancements and addresses some bugs present in earlier Java releases for Mac OS X. Specificially, this update features numerous graphics improvements in the Mac OS X Java implementation. For general information about Java changes in Java 1.4.2_05, see *Release Notes - Java 2 SDK, Standard Edition Version 1.4.2_05* at http://java.sun.com/j2se/1.4.2/ReleaseNotes.html.

How and Where to Get Java 1.4.2 Update 1

Java 1.4.2 Update 1 is available only for Mac OS X version 10.3.4 (Panther) or later. It is not available for earlier versions of Mac OS X. There are two parts to the Java 1.4.2 Update 1 release for Mac OS X: the user and developer tools packages. The user package is available through Mac OS X's Software Update and online at http://www.apple.com/downloads/macosx/. Software Update displays this package as a recommended update.

Apple does not provide a redistribution license for Java 1.4.2 Update 1. Your customers need to download it directly from Apple's site.

The developer tools 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. You need to download this package to access the Java 1.4.2 documentation.

Who Should Read This Document?

Any developer that currently distributes Java 1.4.2 applications for Mac OS X should read this document since the changes to Java 1.4.2 Update 1 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 15) 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 15) chapter. Otherwise it is recommended that Java developers read both 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*
- Previous Java Release Notes
- *Java on Mac OS X Frequently Asked Questions* (http://developer.apple.com/java/faq/)

The following Sun document is especially helpful in determining how new features introduced in Java 1.4.2 Update 1 may affect your applications:

- *Release Notes - Java 2 SDK, Standard Edition Version 1.4.2_05*, available at http://java.sun.com/j2se/1.4.2/ReleaseNotes.html.

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 AWT

Radar #3065640

Drag action modifier key issues.

Description:

There were serveral issues caused by some fundamental mismatches between the Mac OS X drag-and-drop system and the Java drag-and-drop system:

- A `DragSource` was not notified of drag action changes (via `DragSourceListener.dropActionChanged()`) during the drag.
- Once the drag was complete, the `DragSource` was not always notified of the correct final drag action nor was it correctly informed of the success or failure of the drop.
- The `DropTarget` received drag action change notifications, but the notifications were often wrong.

Resolution:

All of the above have now been fixed. The current drag action modifier key behavior is detailed in Table 2-1:

Table 1-1 Drag action modifier key behavior

Modifier Style	Copy	Move	Link	Behavior
Java	Control	Shift	Control+Shift	Within and between Java applications; from native applications to Java applications.
Mac OS X	Option	Command	Command+Option	Within and between Java applications; between native and Java applications.

If you want to determine which action is taking place regardless of the modifier style used, check for `ACTION_COPY`, `ACTION_LINK`, and `ACTION_MOVE` using `DragSourceDragEvent.getUserAction()`.

Radar #3550815

Image background color problems.

Description:

Images would display with a black background.

Resolution:

Images are now contained within components and use the component's background color when drawing.

Radar #3578909

AWT component issues.

Description:

Every time a heavyweight AWT component was added to a container, it would validate and lay out the container.

Resolution:

Adding a component no longer forces the container to validate and lay itself out.

Radar #3730358

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

Description:

Microsoft Office installs fonts into `~/Library/Fonts/`. Once installed, these fonts are used by Java applications instead of those installed with the system. This causes problems with the standard `Dialog` and `Serif` fonts, Arial and Times New Roman.

Resolution:

When conflicting fonts are present, the system-installed copies are used.

Java Events

Radar #2669001

Element dragging displays an oddly-shaped outline.

Description:

Dragging a `JList`, `JTable`, or `JTree` element causes an outline much larger than the element to be displayed during the drag.

Resolution:

Dragging an element now outlines just the element.

Radar #3550412

Right-button drag-and-drop operations disable all further drag-and-drop operations.

Description:

Mac OS X does not support right-button drag-and-drop operations. This caused an `ASSERT` and disabled all drag-and-drop operations for a Java application.

Resolution:

A right-button drag-and-drop operation no longer causes an `ASSERT`. Note that right-button drag-and-drop behavior is still unsupported.

Radar #3571921

DropTargets fail to register as valid.

Description:

Sometimes a `DropTarget` failed to register, causing drag-and-drop operations to fail.

Resolution:

All `DropTarget` components now register properly.

Radar #3593460

Non-serializable objects could not be dragged.

Description:

In some cases, non-serializable objects could not be dragged within an application because they were not properly registered with the Mac OS X drag-and-drop system.

Resolution:

Non-serializable objects can now be dragged within an application.

Radar #3607770

Drag-and-drop coordinate system issues.

Description:

During drag-and-drop, a `DragSourceListener` can be registered to receive notifications about the progress of the drag operation. The notifications contain events that can be queried for the location of the mouse cursor at the time of the event. Previously, you were supplied with the cursor location in window-relative coordinates, even though Java specifies that the cursor location be supplied in screen coordinates.

Resolution:

The mouse cursor location is now returned in screen coordinates.

Java Graphics

Radar #3481676

Full screen mode support.

Description:

Full screen mode was problematic and frequently caused issues.

Resolution:

Full screen mode now functions as expected.

Java HotSpot

Radar #3569904

Long variables and the PowerPC G5.

Description:

Long variables could be put into an unknown state when running a Java application on a G5-equipped machine.

Resolution:

Long variables are now properly handled.

Radar #3644530

MaxPermSize increased.

Description:

Server Java applications were hitting against memory size limitations.

Resolution:

The default `MaxPermSize` for all Java applications has been increased to 64MB.

Radar #3710160

Java profiling agent library locations.

Description:

Java profiling agent libraries had to be installed in `/System/Library/Frameworks/JavaVM.framework/Libraries/`.

Resolution:

Java profiling agent libraries can now be installed in `/System/Library/Java/Extensions/` or `/Library/Java/Extensions/`.

Java Plugin

Radar #3168316

Applet cache is invisible to the user.

Description:

The Java applet cache was stored in `~/.java/deployment/`, which is not visible to the user. If the user was low on disk space and had run a number of applets, the only way to free up space was to run the Java 1.4.2 Plugin Settings application.

Resolution:

The Applet cache has been moved to `~/Library/Caches/Java Applets/`. This change makes it clearer to identify large directories that can safely be removed.

Radar #3505649

Various Java Plugin improvements.

Description:

There were various issues involving the Java Plugin and AWT events, applet sizing, and standard Java applet properties.

Resolution:

The Java Plugin now uses more standard Java Plugin code from Sun when running applets. Due to these changes, AWT Events are now delivered properly. Dynamically sized applets and standard Java applet properties are also supported.

Note that in fixing these issues, these files have been removed:

- `/Library/Internet Plug-Ins/JavaPluginCocoa.bundle/Contents/Resources/AppletsInCocoa.jar`
- `/Library/Internet Plug-Ins/JavaPluginCocoa.bundle/Contents/MacOS/libAppletsInCocoa.jnilib`

Java Security

Radar #3527098

Keystore issues between Java versions.

Description:

A JCE keystore written by the Java 1.4.1 virtual machine (VM) could not be read using the Java 1.4.2 VM.

Resolution:

The Java 1.4.2 VM can now read JCE keystores written by the Java 1.4.1 VM. This is related to Sun bug 4887561, as fixed in the Java 1.4.2_05 VM.

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 #3627635

IntelliJ IDEA needs to upgraded after applying Java 1.4.2 Update 1.

Description:

IntelliJ IDEA version 4.03 works around a bug in previous versions of the Java virtual machine. This bug has been fixed in Java 1.4.2 Update 1.

Workaround:

For IntelliJ IDEA to work properly after applying Java 1.4.2 Update 1, you need to update to version 4.5, available from JetBrains at http://www.jetbrains.com/.

Java Documentation

Radar #3705168

Java developer documentation changes.

Description:

Previously, Java developer documentation needed to be manually decompressed using a script provided by Apple. Running the script resulted in documentation placed in `/Developer/Documentation/Java/`.

Workaround:

Installing the Java 1.4.2 Update 1 Developer Tools package automatically places the Java developer documentation in `/Developer/ADC Reference Library/documentation/Java/`. Because of this change, there is no need to manually invoke the Java developer documentation script, even though it may be present.

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 using the `apple.laf.useScreenMenuBar` property, the method `JMenu.getLocationOnScreen()` returns the location of `JMenu` components as if they were not in the menu bar.

Workaround:

None.

Document Revision History

This table describes the changes to *Java 1.4.2 Update 1 Release Notes*.

Date	Notes
2004-08-11	First version of *Java 1.4.2 Update 1 Release Notes*.

Document Revision History