J2SE 5.0 Release 3 Release Notes

Java

2006-01-10

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

Simultaneously published in the United States and Canada.

Contents

Introduction **Introduction to J2SE 5.0 Release 3 Release Notes 7**

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

Chapter 1 **Using J2SE 5.0 on Mac OS X v.10.4 9**

Java Versioning 9
Java Developer 10

Chapter 2 **Resolved Issues 13**

Java Applets 13
Java Aqua Look and Feel 13
Java AWT 13
Java Events 15
Java Libraries 16
Java Networking 17
Java Printing 17
Java Swing 18
Java SWT Support 18
Java Text 18
Java Virtual Machine 19
Java Web Start 20
Other Resolved Issues 21

Chapter 3 **Outstanding Issues 25**

Java AWT 25
Java Developer 25
Java Security 26
Java Text 26
Java Web Start 26

Document Revision History 27

Tables

Chapter 1 **Using J2SE 5.0 on Mac OS X v.10.4 9**

Table 1-1 Java virtual machine behavior, based on the JVMVersion key and the Java Version Precedence List in Java Preferences 10

Chapter 2 **Resolved Issues 13**

Table 2-1 Various Resolved Issues in J2SE 5.0 Release 3 21

Introduction to J2SE 5.0 Release 3 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_05.

What is J2SE 5.0 Release 3 for Mac OS X?

J2SE 5.0 Release 3 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_05).

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 3, Java 1.4.2 remains the default version of Java used for applications and applets unless J2SE 5.0 is specifically requested by the application or configured for use system-wide. For more on using J2SE 5.0, consult "Using J2SE 5.0 on Mac OS X v.10.4" (page 9).

Who Should Read This Document?

Any developer who wants to distribute J2SE 5.0 applications for Mac OS X should read this document since various issues and fixes found in this release may effect 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:

- "Using J2SE 5.0 on Mac OS X v.10.4" (page 9) discusses how to specify J2SE 5.0 as your preferred version of Java for applications and applets. It also includes information on using J2SE 5.0 from within Xcode, the integrated development environment from Apple.
- "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 25) 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 25) 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/)

Using J2SE 5.0 on Mac OS X v.10.4

This chapter provides information on packaging your application so that it can require J2SE 5.0. It also includes the steps you need to take when developing J2SE 5.0 applications on Mac OS X v.10.4.

Java Versioning

Radar #4030615

The default version of Java after installing J2SE 5.0 Release 3

Description:

Java 1.4.2 remains the default version of Java used by applications and applets after installing J2SE 5.0 Release 3.

Resolution:

This allows applications built for Java 1.4.2 to continue running in Java 1.4.2.

To use the J2SE 5.0 version of any command-line Java utility, such as `java` or `javac`, specify the full path to the J2SE 5.0 version of the command. This usually means prepending the command with the path `/System/Library/Frameworks/JavaVM.framework/Versions/1.5/Commands/`.

The `JVMVersion` Info.plist key allows you to specify which Java virtual machine your application uses. Using either the 1.3+, 1.4+, or 1.5+ keys results in the Java virtual machine being used according to a precedence list set in the Java Application Settings portion of the Java Preferences application, found in `/Applications/Utilities/Java/J2SE 5.0`. By default in J2SE 5.0 Release 3, the precedence list follows this order:

```
J2SE 1.4.2
J2SE 5.0
```

Though not listed, Java 1.3.1 is considered to be the last entry on the list.

This works by trying to match the given key versus Java virtual machine versions in the list:

- Having 1.4+ specified first matches the "J2SE 1.4.2" value, so that is the Java virtual machine used. Specifying 1.5+ means that 1.4.2 is skipped and the next choice, "J2SE 5.0," is used.
- If a user switches these values so that "J2SE 5.0" is first, an application that specifies 1.4+ runs in J2SE 5.0, since it's the first entry and numerically superior to Java 1.4.2.

Table 1-1 (page 10) lists the possible combinations of values when working with `JVMVersion` values and the Java Version Precedence List found in Java Preferences.

Table 1-1 Java virtual machine behavior, based on the JVMVersion key and the Java Version Precedence List in Java Preferences

Possible JVMVersion Value	Java virtual machine used if "J2SE 1.4.2" is listed first in Java Version Precedence List (Default)	Java virtual machine used if "J2SE 5.0" is listed first in Java Version Precedence List
`1.5+`*(5.0 or "higher")*	J2SE 5.0	J2SE 5.0
`1.5*`*(newest 5.0)*	J2SE 5.0	J2SE 5.0
`1.4+`*(1.4 or "higher")*	Java 1.4.2	J2SE 5.0
`1.4*`*(newest 1.4)*	Java 1.4.2	Java 1.4.2
`1.3+`*(1.3 or "higher")*	Java 1.4.2	J2SE 5.0
`1.3*`*(newest 1.3)*	Java 1.3.1	Java 1.3.1

You can specify a specific version of Java as your `JVMVersion` key (such as *1.4.2* or *1.5.0*), but if that precise version is not installed on the system, your application will fail to launch. *This is not recommended.*

To use an applet in Safari using J2SE 5.0, use the Applet portion of the Java Preferences application to select J2SE 5.0 as the virtual machine for applets. Java Preferences is found in `/Applications/Utilities/Java/J2SE 5.0`,

For information on configuring Xcode projects for use with J2SE 5.0, read "Xcode Java Projects in J2SE 5.0" (page 10).

Java Developer

Radar #4090917

Xcode Java Projects in J2SE 5.0

Description:

The Java project templates in Xcode are set up for use with Java 1.4.2.

Workaround:

Follow these steps to modify projects for use with J2SE 5.0:

Target Settings:

Double click the target to edit and provide `/System/Library/Frameworks/JavaVM.framework/Versions/1.5/Commands/javac` as the value for the `JAVA_COMPILER` build setting, available in the Expert View. Change the `Target VM Version` and `Source Version` in the `Java Compiler Setting` to use `1.5`.

Executable Settings:

Double click the executable named `java` and enter `/System/Library/Frameworks/JavaVM.framework/Versions/1.5/Commands/java` as the `Executable Path` in the *General* tab of Executable info.

Applet Development:

Double click the executable named `appletviewer` and enter `/System/Library/Frameworks/JavaVM.framework/Versions/1.5/Commands/appletviewer` as the `Executable Path` in the *General* tab of Executable info.

The Java Xcode project templates are currently set to specify a `JVMVersion` of `1.4*`, meaning that your compiled application runs in Java 1.4.2. To run your application in J2SE 5.0, set the `Target VM Version` to `1.5*`:

1. Open the Target window for your application's build target
2. Select the `Pure Java Specific` listing under `Info.plist Entries`
3. For the `Target VM Version` field, enter `1.5*`

Radar #4090919

J2SE 5.0 Reference Documentation

Description:

The J2SE 5.0 Reference Documentation is not installed automatically.

Workaround:

Install the J2SE 5.0 Release 3 Documentation package, available from the Downloads > Java page at http://connect.apple.com.

Note: J2SE 5.0 symbols are not used in the Xcode Documentation window by default. To use the 5.0 symbols instead of the 1.4.2 symbols, you need to modify which version of Java is indexed. To do this, follow the directions outlined in the workaround for "J2SE 5.0 symbols not visible in Xcode Documentation window" (page 11).

Radar #4310936

J2SE 5.0 symbols not visible in Xcode Documentation window

Description:

After installing the J2SE 5.0 Reference documentation or a documentation update in Xcode, J2SE 5.0 symbols are not present in the Xcode documentation window.

Workaround:

You need to manually run the `pbhelpindexer` utility for the J2SE 5.0 symbols to appear in the Xcode Documentation window. First, open `MacOSXDeveloper.pbHelpIndexerList`, located at:

```
/Developer/ADC Reference Library/indexes/
```

Replace this path:

```
/System/Library/Frameworks/JavaVM.framework/Versions/CurrentJDK/Resources/
```

With this path:

```
/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Resources/
```

Finally, run this command in Terminal (it may take a while to finish executing):

```
sudo /Developer/Tools/pbhelpindexer
```

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 #4216053

LiveConnect and Dialogs

Description:

Attempting to show a dialog, like a `FileDialog`, using LiveConnect caused a deadlock that could hang the host application.

Resolution:

Showing a dialog via LiveConnect no longer causes a deadlock.

Java Aqua Look and Feel

Radar #4266079

JProgressBar Prevents Application Exit

Description:

The animation timer used in painting the Aqua progress bar did not stop when the progress bar was hidden. This could prevent the component from disposing.

Resolution:

The animation timer is now halted, allowing for proper component disposal.

Java AWT

Radar #4008177

MouseInfo.getNumberOfButtons() Return Values

Description:

The method `MouseInfo.getNumberOfButtons()` always returned a value of `0`, regardless of the number of buttons on the mouse attached to the system.

Resolution:

The correct number of mouse buttons are now returned. If no mouse is attached, -1 is returned.

Radar #4156678

Headless Applications and AWT

Description:

In a headless application, trying to use any AWT class caused a debugging message to be printed, halting the application while waiting for user input.

Resolution:

The debugging message no longer appears, allowing the application to continue executing.

Radar #4160457

Pasteboard Keyboard Shortcuts and Swing Applets

Description:

Swing applets weren't receiving copy, cut, and paste events that were the result of a keyboard shortcut (Command-C, Command-X, and Command-V, respectively).

Resolution:

The keyboard shortcuts for Pasteboard operations now function as expected. Note that Swing applets have their own pasteboards that are not shared with the system's general pasteboard.

Radar #4164891

Window Memory Leak

Description:

Some Window and Window subclass objects leaked memory.

Resolution:

Disposed Windows no longer leak memory.

Radar #4175954

Robot.createScreenCapture Crash

Description:

Calling `Robot.createScreenCapture()` with invalid bounds (larger than the actual display) caused the application to crash or the return a corrupted image.

Resolution:

The application no longer crashes. Values outside of the display are black in the returned image.

Radar #4201970

Nested Modal Dialogs

Description:

Nested modal dialogs don't behave properly if they all use a common parent. For instance, a focused `Dialog` may end up behind an unfocused `Dialog`.

Resolution:

Nested modal `Dialogs` maintain their ordering.

Radar #4247594

Graphical Java applications and Case-sensitive Filesystems

Description:

On case-sensitive filesystems, like UFS, Java applications that try to show a user interface failed to launch.

Resolution:

Graphical Java applications launch normally on all filesystems.

Java Events

Radar #4086462

Malformed MouseEvents

Description:

When the Command key is held down, a `MouseEvent` sometimes had an incorrect source Component. When the Command key was held down, a `MouseEvent` sometimes reported that the third mouse button was down when it was not. When the Command key was held down, `MOUSE_CLICKED` events were not delivered to unfocused Components. When the Shift key was held down, a `MouseEvent` sometimes reported that the first mouse button was down when it was not.

Resolution:

`MouseEvents` are no longer malformed when Command or Shift are held down.

Radar #4176668

Robot Mouse Events

Description:

Because Robot was set to take precedence over genuine user events, posting a `mousePress` without later posting a `mouseRelease` caused the entire user interface to stop responding to mouse clicks.

Resolution:

Robot events no longer take precedence over genuine user events.

Radar #4237416

Custom DataFlavors Across Separate Virtual Machines

Description:

If a Java process defined a custom `DataFlavor`, other Java processes could not read data stored within that DataFlavor.

Resolution:

A custom `DataFlavor` is now read properly by the general system clipboard.

Radar #4238470

Images on the System Pasteboard

Description:

When an image was copied to the general system pasteboard, the image was corrupted.

Resolution:

The image is no longer corrupted.

Java Libraries

Radar #4084603

JMX and jconsole Fixes

Description:

The `management.properties` file wasn't available on J2SE 5.0 Release 1, disabling JMX features.

Resolution:

The `management.properties` file is available with this release. To work with JMX, `jconsole` has been updated to connect correctly to the virtual machine.

Radar #4116981

Compile Failure with .zip or .jar File Extensions

Description:

If the classpath had entries that pointed to files without `.zip` or `.jar` extensions, `javac` issued a warning about the file and refused to compile the executable. This bug is found in Sun's bug database as Bug ID 6295519.

Resolution:

`javac` ignores invalid files with `.zip` and `.jar` extensions.

Radar #4149837

Reopening Files with java.io.RandomAccessFile

Description:

When a file was created with `rw` permissions and then shortly thereafter a new `RandomAccessFile` object was instantiated for the same file in `rws` mode, a `FileNotFoundException (File Exists)` exception is thrown.

Resolution:

`RandomAccessFile` no longer throws an exception in this case.

Java Networking

Radar #4236458

Reusing Addresses with DatagramChannels

Description:

Calling `setReuseAddress()` on a `DatagramChannel` did not have any effect.

Resolution:

Calling `setReuseAddress()` now reuses the address.

Java Printing

Radar #4093580

Using Multiple Printers

Description:

Print jobs were often printed to the default printer instead of the selected printer.

Resolution:

A targeted printer prints the document, as expected.

Java Swing

Radar #4134772

Swapping Menu Bars Within the Screen Menu Bar

Description:

When an application used the screen menu bar and swapped between multiple menu bars via `setJMenuBar`, duplicate menus were shown.

Resolution:

Duplicate menus are no longer shown.

Java SWT Support

Radar #4091298

SWT Applications Using AWT Classes

Description:

Using an AWT class within a SWT application is normally not possible, due to issues with threading during the startup of the AWT. One workaround was to set the `-Djava.awt.headless` flag equal to `true` at runtime.

Resolution:

AWT startup detects the existence of the SWT and sets up AWT in headless mode automatically. Using the `-Djava.awt.headless` is no longer necessary.

Java Text

Radar #4122177

Distorted Text Layout Bound Values

Description:

The bounds returned by `java.awt.font.TextLayout.getBounds()` were inaccurate.

Resolution:

Returned values are more accurate.

Radar #4164128

Incorrect FontMetrics in a Scaled Graphics2D Object

Description:

When a `Graphics2D` object was scaled using its `scale` method, the text dimensions as reported by a `FontMetrics` object were incorrect.

Resolution:

The reported text dimensions are now correct.

Radar #4176164

Transformed Strings

Description:

Scaled or rotated strings were drawn with incorrect spacing between the letters, doubly transformed glyphs, or glyphs scaled non-proportionally.

Resolution:

Scaled and rotated strings now draw correctly.

Radar #4156757

Display Issues after Character Set Change

Description:

Changing the input source character set wasn't immediately respected by input fields.

Resolution:

Input source character set changes take effect immediately.

Java Virtual Machine

Radar #3499564

JVM Stat Tools

Description:

The jvmstat tools were not available with J2SE 5.0 Release 1.

Resolution:

`/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Commands/jps`
Prints the process ID (`<pid>`) for all active Java processes

`/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Commands/jinfo -flags <pid>`
Prints the VM flags for the specified process

`/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Commands/jinfo -sysprops <pid>`

Prints the Java System properties for the specified process

`/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Commands/jmap -heap <pid>`

Prints the Java Heap summary for the specified process

`/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Commands/jmap -heap:format=b <pid>`

Prints the Java Heap for the specified process in the hprof binary format to a file

`/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Commands/jmap -histo <pid>`

Prints a histogram of Java object heap for the specified process

`/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Commands/jmap -permstat <pid>`

Prints classloader-wise permanent generation statistics for the process

`/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Commands/jstack <pid>`

Prints the Java stack dump

For more information go to:

- http://java.sun.com/performance/jvmstat/
- http://java.sun.com/j2se/1.5.0/docs/tooldocs/

Java Web Start

Radar #4074412

Ignored Descriptor Elements

Description:

Java Cache Viewer didn't support JNLP documents that had an "association" element in their "information" descriptor. The application worked as desired, but any declared associations were not registered with Launch Services when the user created a desktop application.

Resolution:

An application's "association" element is registered properly.

Radar #4088809

Saving Web Start Applications as Local Applications

Description:

Launching a local JNLP application after it was modified caused the Save dialog to appear again.

Resolution:

Updated applications no longer need to be saved again.

Radar #4126743

Creating Local Applications with Java Cache Viewer

Description:

Java Cache Viewer created local applications that always tried to launch in Java 1.3.1 or would not launch at all.

Workaround:

Java Cache Viewer creates correctly formed desktop applications

Other Resolved Issues

Table 2-1 (page 21) lists numerous issues present in previous versions of Java for Mac OS X that are resolved in J2SE 5.0 Release 3.

Table 2-1 Various Resolved Issues in J2SE 5.0 Release 3

Radar Number	Description
4162239	Host application hung while loading applet
4197144	Host application crashed when resizing during applet load
4275445	Host application hung when resizing or zooming a window during applet load
4295651	Host application crashed while reloading applets
3993081	Command key and mouse event incorrectly deselected the current combo box
4164430	Variable `JTableHeader` heights displayed improperly
4165961	Extraneous Open button used in `JFileChooser`
4176694	Navigating a `JFileChooser` to a shared volume caused a crash
4202729	Visual delineation between words in locales such as Japanese were missing
4251968	Null Pointer Exception in `ScreenPopupFactory`
4273831	Minimizing internal frames threw an exception
3132190	Components were painted twice when first shown
3155258	AWT Scrollbars didn't draw properly when resized or moved
4127577	Popup menus appeared on different monitors than their associated controls
3310174	`apple.awt.window.position.forceSafeProgrammaticPositioning` forced windows onto the primary display
3748153	Scrollwheels could not be used within popup menus

Radar Number	Description
4073021	Forward-delete performed a standard delete
4120464	Crash when `TextArea.replaceRange()` was called with invalid values
4127790	Fonts for AWT Components were wrong
4245892	Jar files packed using pack200 were corrupt
4129065	Windows flickered when adding a heavyweight component to a Container
4133696	Calling `TextField.getText()` caused repaint
4156683	Validating a container was slow
4156692	Applet background changes a host window's background
4166324	String leak when using Input Methods
4170826	Systems colors returned from the `UIManager` were incorrect
4172490	Showing a Dialog with a minimized parent caused a "phantom" window to appear
4176931	Live Resize could cause a deadlock
3618611	An image with a negative width or height was not reversed
4160173	`ImageIO.read()` swapped color channels
4181693	AWT Lists were unresponsive
4274285	`Component.requestFocusInWindow()` didn't always set key focus
4184499	`TextArea.setText()` didn't always set text area's text
4185651	Applications crashed when dismissing `JColorChooser` or `JFileChooser`
4187162	Submenus were not accessible from a modal dialog
4197676	Menus displayed at the wrong vertical position when using multiple monitors
4217124	`setUncaughtExceptionHandler` didn't work for exceptions thrown from the AWT Event dispatch thread
4304193	`Container.setComponentZOrder()` didn't update the user interface
4137834	`DragSource.dragEnter()` and `DragSource.dragExit()` event were sent at the wrong times
3843735	Shift-backspace wasn't treated like backspace for Swing Text controls
4156798	Tablets generated unwanted console messages
4161644	When Shift key was held down, `MouseWheelEvents` were incorrectly reported

Radar Number	Description
4224122	Enter didn't insert a carriage return when typing in Korean
3955254	Some menu accelerator keys were processed as `ActionEvents`, not `KeyEvents`
3824226	Applets could draw outside their bounds
4151808	Images of `TYPE_INT_RGB` rendered slowly
4156466	Graphics didn't update properly in applet subwindows
4160169	xRGB images were not displayed correctly
4161593	Textures weren't printed properly
4097397	The virtual machine didn't allow the "MB" stack memory size modifier
4106727	Applications crashed during garbage collection
4190619	The Java virtual machine didn't warn when Train garbage collector was requested
4255144	Better compatibility with profiling utilities
4096926	The `net.properties` file was missing
4160100	The `dnsns.jar` file was missing
4164912	Memory leak when printing
4203055	Window ordering problems with combo boxes and `AlwaysOnTop` windows
4107761	Applications crashed on launch when Courier or Times fonts were missing
4260764	Applications crashed if Helvetica was missing
4132856	Scaled fonts using integer metrics exhibited an odd spacing between characters
4270646	Certificate Trust dialog defaulted to "Yes"
4158179	Desktop applications could not be created from a launched JNLP file
4085734	Java Cache Viewer was missing a "Help" menu

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 AWT

Radar #4307013

Focus Issues

Description:

Focus issues with this release include:

- Switching focus away from an AWT `TextArea` to a non-focusable component (such as `Canvas`) and back to the `TextArea` causes focus to be permanently lost
- Switching focus away from a Swing `JTextArea` to an AWT heavyweight component (such as a `Panel`) and back again causes focus to be permanently lost

Workaround:

None.

Java Developer

Radar #4319434

J2SE 5.0 Symbols in Xcode Documentation Window

Description:

The `pbhelpindexer` command indexes the Java Reference Documentation and provides symbols for use with the Xcode Documentation window. The version included with Xcode 2.0 and 2.1 didn't support some of the new constructs available in J2SE 5.0.

Workaround:

Upgrade to Xcode 2.2 and use the version of `phhelpindexer` that's included with it.

Java Security

Radar #4115657

Keys in the KeychainStore

Description:

The `KeychainStore` cannot create a `Key` object from private keys stored in the user's Keychain. Keys can be enumerated and you can view the certificate chain associated with the key, but you can't retrieve a `Key` object.

Workaround:

None.

Java Text

Radar #4337704

JTextField in Metal or Synth Look and Feel

Description:

Text in a `JTextField` may have slight graphical problems when using the Metal or Synth look and feel. It is particularly noticeable when selecting text.

Workaround:

Use the `AlwaysUseCoreText` runtime value:

```
-Dapple.awt.AlwaysUseCoreText=true
```

Java Web Start

Radar #4075884

JNLP Memory Settings and Property Values

Description:

A JNLP file can specify a custom memory setting or a property that would normally be set from the command line, like `apple.laf.useScreenMenuBar`. That setting, however, is not copied into the `Java` dictionary of the `Info.plist` when a user creates a local application.

Workaround:

None.

Document Revision History

This table describes the changes to *J2SE 5.0 Release 3 Release Notes*.

Date	Notes
2006-01-10	Fixed typos throughout the document.
2005-11-15	First draft of J2SE 5.0 Release 3 Release Notes

Document Revision History