| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV PACKAGE**](http://docs.google.com/java/applet/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/awt/color/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package java.awt

Contains all of the classes for creating user interfaces and for painting graphics and images.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**ActiveEvent**](http://docs.google.com/java/awt/ActiveEvent.html) | An interface for events that know how to dispatch themselves. |
| [**Adjustable**](http://docs.google.com/java/awt/Adjustable.html) | The interface for objects which have an adjustable numeric value contained within a bounded range of values. |
| [**Composite**](http://docs.google.com/java/awt/Composite.html) | The Composite interface, along with [CompositeContext](http://docs.google.com/java/awt/CompositeContext.html), defines the methods to compose a draw primitive with the underlying graphics area. |
| [**CompositeContext**](http://docs.google.com/java/awt/CompositeContext.html) | The CompositeContext interface defines the encapsulated and optimized environment for a compositing operation. |
| [**ItemSelectable**](http://docs.google.com/java/awt/ItemSelectable.html) | The interface for objects which contain a set of items for which zero or more can be selected. |
| [**KeyEventDispatcher**](http://docs.google.com/java/awt/KeyEventDispatcher.html) | A KeyEventDispatcher cooperates with the current KeyboardFocusManager in the targeting and dispatching of all KeyEvents. |
| [**KeyEventPostProcessor**](http://docs.google.com/java/awt/KeyEventPostProcessor.html) | A KeyEventPostProcessor cooperates with the current KeyboardFocusManager in the final resolution of all unconsumed KeyEvents. |
| [**LayoutManager**](http://docs.google.com/java/awt/LayoutManager.html) | Defines the interface for classes that know how to lay out Containers. |
| [**LayoutManager2**](http://docs.google.com/java/awt/LayoutManager2.html) | Defines an interface for classes that know how to layout Containers based on a layout constraints object. |
| [**MenuContainer**](http://docs.google.com/java/awt/MenuContainer.html) | The super class of all menu related containers. |
| [**Paint**](http://docs.google.com/java/awt/Paint.html) | This Paint interface defines how color patterns can be generated for [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html) operations. |
| [**PaintContext**](http://docs.google.com/java/awt/PaintContext.html) | The PaintContext interface defines the encapsulated and optimized environment to generate color patterns in device space for fill or stroke operations on a [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html). |
| [**PrintGraphics**](http://docs.google.com/java/awt/PrintGraphics.html) | An abstract class which provides a print graphics context for a page. |
| [**Shape**](http://docs.google.com/java/awt/Shape.html) | The Shape interface provides definitions for objects that represent some form of geometric shape. |
| [**Stroke**](http://docs.google.com/java/awt/Stroke.html) | The Stroke interface allows a [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html) object to obtain a [Shape](http://docs.google.com/java/awt/Shape.html) that is the decorated outline, or stylistic representation of the outline, of the specified Shape. |
| [**Transparency**](http://docs.google.com/java/awt/Transparency.html) | The Transparency interface defines the common transparency modes for implementing classes. |

| **Class Summary** | |
| --- | --- |
| [**AlphaComposite**](http://docs.google.com/java/awt/AlphaComposite.html) | The AlphaComposite class implements basic alpha compositing rules for combining source and destination colors to achieve blending and transparency effects with graphics and images. |
| [**AWTEvent**](http://docs.google.com/java/awt/AWTEvent.html) | The root event class for all AWT events. |
| [**AWTEventMulticaster**](http://docs.google.com/java/awt/AWTEventMulticaster.html) | AWTEventMulticaster implements efficient and thread-safe multi-cast event dispatching for the AWT events defined in the java.awt.event package. |
| [**AWTKeyStroke**](http://docs.google.com/java/awt/AWTKeyStroke.html) | An AWTKeyStroke represents a key action on the keyboard, or equivalent input device. |
| [**AWTPermission**](http://docs.google.com/java/awt/AWTPermission.html) | This class is for AWT permissions. |
| [**BasicStroke**](http://docs.google.com/java/awt/BasicStroke.html) | The BasicStroke class defines a basic set of rendering attributes for the outlines of graphics primitives, which are rendered with a [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html) object that has its Stroke attribute set to this BasicStroke. |
| [**BorderLayout**](http://docs.google.com/java/awt/BorderLayout.html) | A border layout lays out a container, arranging and resizing its components to fit in five regions: north, south, east, west, and center. |
| [**BufferCapabilities**](http://docs.google.com/java/awt/BufferCapabilities.html) | Capabilities and properties of buffers. |
| [**BufferCapabilities.FlipContents**](http://docs.google.com/java/awt/BufferCapabilities.FlipContents.html) | A type-safe enumeration of the possible back buffer contents after page-flipping |
| [**Button**](http://docs.google.com/java/awt/Button.html) | This class creates a labeled button. |
| [**Canvas**](http://docs.google.com/java/awt/Canvas.html) | A Canvas component represents a blank rectangular area of the screen onto which the application can draw or from which the application can trap input events from the user. |
| [**CardLayout**](http://docs.google.com/java/awt/CardLayout.html) | A CardLayout object is a layout manager for a container. |
| [**Checkbox**](http://docs.google.com/java/awt/Checkbox.html) | A check box is a graphical component that can be in either an "on" (true) or "off" (false) state. |
| [**CheckboxGroup**](http://docs.google.com/java/awt/CheckboxGroup.html) | The CheckboxGroup class is used to group together a set of Checkbox buttons. |
| [**CheckboxMenuItem**](http://docs.google.com/java/awt/CheckboxMenuItem.html) | This class represents a check box that can be included in a menu. |
| [**Choice**](http://docs.google.com/java/awt/Choice.html) | The Choice class presents a pop-up menu of choices. |
| [**Color**](http://docs.google.com/java/awt/Color.html) | The Color class is used to encapsulate colors in the default sRGB color space or colors in arbitrary color spaces identified by a [ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html). |
| [**Component**](http://docs.google.com/java/awt/Component.html) | A *component* is an object having a graphical representation that can be displayed on the screen and that can interact with the user. |
| [**ComponentOrientation**](http://docs.google.com/java/awt/ComponentOrientation.html) | The ComponentOrientation class encapsulates the language-sensitive orientation that is to be used to order the elements of a component or of text. |
| [**Container**](http://docs.google.com/java/awt/Container.html) | A generic Abstract Window Toolkit(AWT) container object is a component that can contain other AWT components. |
| [**ContainerOrderFocusTraversalPolicy**](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html) | A FocusTraversalPolicy that determines traversal order based on the order of child Components in a Container. |
| [**Cursor**](http://docs.google.com/java/awt/Cursor.html) | A class to encapsulate the bitmap representation of the mouse cursor. |
| [**DefaultFocusTraversalPolicy**](http://docs.google.com/java/awt/DefaultFocusTraversalPolicy.html) | A FocusTraversalPolicy that determines traversal order based on the order of child Components in a Container. |
| [**DefaultKeyboardFocusManager**](http://docs.google.com/java/awt/DefaultKeyboardFocusManager.html) | The default KeyboardFocusManager for AWT applications. |
| [**Desktop**](http://docs.google.com/java/awt/Desktop.html) | The Desktop class allows a Java application to launch associated applications registered on the native desktop to handle a [URI](http://docs.google.com/java/net/URI.html) or a file. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html) | A Dialog is a top-level window with a title and a border that is typically used to take some form of input from the user. |
| [**Dimension**](http://docs.google.com/java/awt/Dimension.html) | The Dimension class encapsulates the width and height of a component (in integer precision) in a single object. |
| [**DisplayMode**](http://docs.google.com/java/awt/DisplayMode.html) | The DisplayMode class encapsulates the bit depth, height, width, and refresh rate of a GraphicsDevice. |
| [**Event**](http://docs.google.com/java/awt/Event.html) | **NOTE:** The Event class is obsolete and is available only for backwards compatilibility. |
| [**EventQueue**](http://docs.google.com/java/awt/EventQueue.html) | EventQueue is a platform-independent class that queues events, both from the underlying peer classes and from trusted application classes. |
| [**FileDialog**](http://docs.google.com/java/awt/FileDialog.html) | The FileDialog class displays a dialog window from which the user can select a file. |
| [**FlowLayout**](http://docs.google.com/java/awt/FlowLayout.html) | A flow layout arranges components in a directional flow, much like lines of text in a paragraph. |
| [**FocusTraversalPolicy**](http://docs.google.com/java/awt/FocusTraversalPolicy.html) | A FocusTraversalPolicy defines the order in which Components with a particular focus cycle root are traversed. |
| [**Font**](http://docs.google.com/java/awt/Font.html) | The Font class represents fonts, which are used to render text in a visible way. |
| [**FontMetrics**](http://docs.google.com/java/awt/FontMetrics.html) | The FontMetrics class defines a font metrics object, which encapsulates information about the rendering of a particular font on a particular screen. |
| [**Frame**](http://docs.google.com/java/awt/Frame.html) | A Frame is a top-level window with a title and a border. |
| [**GradientPaint**](http://docs.google.com/java/awt/GradientPaint.html) | The GradientPaint class provides a way to fill a [Shape](http://docs.google.com/java/awt/Shape.html) with a linear color gradient pattern. |
| [**Graphics**](http://docs.google.com/java/awt/Graphics.html) | The Graphics class is the abstract base class for all graphics contexts that allow an application to draw onto components that are realized on various devices, as well as onto off-screen images. |
| [**Graphics2D**](http://docs.google.com/java/awt/Graphics2D.html) | This Graphics2D class extends the [Graphics](http://docs.google.com/java/awt/Graphics.html) class to provide more sophisticated control over geometry, coordinate transformations, color management, and text layout. |
| [**GraphicsConfigTemplate**](http://docs.google.com/java/awt/GraphicsConfigTemplate.html) | The GraphicsConfigTemplate class is used to obtain a valid [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html). |
| [**GraphicsConfiguration**](http://docs.google.com/java/awt/GraphicsConfiguration.html) | The GraphicsConfiguration class describes the characteristics of a graphics destination such as a printer or monitor. |
| [**GraphicsDevice**](http://docs.google.com/java/awt/GraphicsDevice.html) | The GraphicsDevice class describes the graphics devices that might be available in a particular graphics environment. |
| [**GraphicsEnvironment**](http://docs.google.com/java/awt/GraphicsEnvironment.html) | The GraphicsEnvironment class describes the collection of [GraphicsDevice](http://docs.google.com/java/awt/GraphicsDevice.html) objects and [Font](http://docs.google.com/java/awt/Font.html) objects available to a Java(tm) application on a particular platform. |
| [**GridBagConstraints**](http://docs.google.com/java/awt/GridBagConstraints.html) | The GridBagConstraints class specifies constraints for components that are laid out using the GridBagLayout class. |
| [**GridBagLayout**](http://docs.google.com/java/awt/GridBagLayout.html) | The GridBagLayout class is a flexible layout manager that aligns components vertically, horizontally or along their baseline without requiring that the components be of the same size. |
| [**GridBagLayoutInfo**](http://docs.google.com/java/awt/GridBagLayoutInfo.html) | The GridBagLayoutInfo is an utility class for GridBagLayout layout manager. |
| [**GridLayout**](http://docs.google.com/java/awt/GridLayout.html) | The GridLayout class is a layout manager that lays out a container's components in a rectangular grid. |
| [**Image**](http://docs.google.com/java/awt/Image.html) | The abstract class Image is the superclass of all classes that represent graphical images. |
| [**ImageCapabilities**](http://docs.google.com/java/awt/ImageCapabilities.html) | Capabilities and properties of images. |
| [**Insets**](http://docs.google.com/java/awt/Insets.html) | An Insets object is a representation of the borders of a container. |
| [**JobAttributes**](http://docs.google.com/java/awt/JobAttributes.html) | A set of attributes which control a print job. |
| [**JobAttributes.DefaultSelectionType**](http://docs.google.com/java/awt/JobAttributes.DefaultSelectionType.html) | A type-safe enumeration of possible default selection states. |
| [**JobAttributes.DestinationType**](http://docs.google.com/java/awt/JobAttributes.DestinationType.html) | A type-safe enumeration of possible job destinations. |
| [**JobAttributes.DialogType**](http://docs.google.com/java/awt/JobAttributes.DialogType.html) | A type-safe enumeration of possible dialogs to display to the user. |
| [**JobAttributes.MultipleDocumentHandlingType**](http://docs.google.com/java/awt/JobAttributes.MultipleDocumentHandlingType.html) | A type-safe enumeration of possible multiple copy handling states. |
| [**JobAttributes.SidesType**](http://docs.google.com/java/awt/JobAttributes.SidesType.html) | A type-safe enumeration of possible multi-page impositions. |
| [**KeyboardFocusManager**](http://docs.google.com/java/awt/KeyboardFocusManager.html) | The KeyboardFocusManager is responsible for managing the active and focused Windows, and the current focus owner. |
| [**Label**](http://docs.google.com/java/awt/Label.html) | A Label object is a component for placing text in a container. |
| [**LinearGradientPaint**](http://docs.google.com/java/awt/LinearGradientPaint.html) | The LinearGradientPaint class provides a way to fill a [Shape](http://docs.google.com/java/awt/Shape.html) with a linear color gradient pattern. |
| [**List**](http://docs.google.com/java/awt/List.html) | The List component presents the user with a scrolling list of text items. |
| [**MediaTracker**](http://docs.google.com/java/awt/MediaTracker.html) | The MediaTracker class is a utility class to track the status of a number of media objects. |
| [**Menu**](http://docs.google.com/java/awt/Menu.html) | A Menu object is a pull-down menu component that is deployed from a menu bar. |
| [**MenuBar**](http://docs.google.com/java/awt/MenuBar.html) | The MenuBar class encapsulates the platform's concept of a menu bar bound to a frame. |
| [**MenuComponent**](http://docs.google.com/java/awt/MenuComponent.html) | The abstract class MenuComponent is the superclass of all menu-related components. |
| [**MenuItem**](http://docs.google.com/java/awt/MenuItem.html) | All items in a menu must belong to the class MenuItem, or one of its subclasses. |
| [**MenuShortcut**](http://docs.google.com/java/awt/MenuShortcut.html) | The MenuShortcutclass represents a keyboard accelerator for a MenuItem. |
| [**MouseInfo**](http://docs.google.com/java/awt/MouseInfo.html) | MouseInfo provides methods for getting information about the mouse, such as mouse pointer location and the number of mouse buttons. |
| [**MultipleGradientPaint**](http://docs.google.com/java/awt/MultipleGradientPaint.html) | This is the superclass for Paints which use a multiple color gradient to fill in their raster. |
| [**PageAttributes**](http://docs.google.com/java/awt/PageAttributes.html) | A set of attributes which control the output of a printed page. |
| [**PageAttributes.ColorType**](http://docs.google.com/java/awt/PageAttributes.ColorType.html) | A type-safe enumeration of possible color states. |
| [**PageAttributes.MediaType**](http://docs.google.com/java/awt/PageAttributes.MediaType.html) | A type-safe enumeration of possible paper sizes. |
| [**PageAttributes.OrientationRequestedType**](http://docs.google.com/java/awt/PageAttributes.OrientationRequestedType.html) | A type-safe enumeration of possible orientations. |
| [**PageAttributes.OriginType**](http://docs.google.com/java/awt/PageAttributes.OriginType.html) | A type-safe enumeration of possible origins. |
| [**PageAttributes.PrintQualityType**](http://docs.google.com/java/awt/PageAttributes.PrintQualityType.html) | A type-safe enumeration of possible print qualities. |
| [**Panel**](http://docs.google.com/java/awt/Panel.html) | Panel is the simplest container class. |
| [**Point**](http://docs.google.com/java/awt/Point.html) | A point representing a location in (x,y) coordinate space, specified in integer precision. |
| [**PointerInfo**](http://docs.google.com/java/awt/PointerInfo.html) | A class that describes the pointer position. |
| [**Polygon**](http://docs.google.com/java/awt/Polygon.html) | The Polygon class encapsulates a description of a closed, two-dimensional region within a coordinate space. |
| [**PopupMenu**](http://docs.google.com/java/awt/PopupMenu.html) | A class that implements a menu which can be dynamically popped up at a specified position within a component. |
| [**PrintJob**](http://docs.google.com/java/awt/PrintJob.html) | An abstract class which initiates and executes a print job. |
| [**RadialGradientPaint**](http://docs.google.com/java/awt/RadialGradientPaint.html) | The RadialGradientPaint class provides a way to fill a shape with a circular radial color gradient pattern. |
| [**Rectangle**](http://docs.google.com/java/awt/Rectangle.html) | A Rectangle specifies an area in a coordinate space that is enclosed by the Rectangle object's upper-left point (x,y) in the coordinate space, its width, and its height. |
| [**RenderingHints**](http://docs.google.com/java/awt/RenderingHints.html) | The RenderingHints class defines and manages collections of keys and associated values which allow an application to provide input into the choice of algorithms used by other classes which perform rendering and image manipulation services. |
| [**RenderingHints.Key**](http://docs.google.com/java/awt/RenderingHints.Key.html) | Defines the base type of all keys used along with the [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) class to control various algorithm choices in the rendering and imaging pipelines. |
| [**Robot**](http://docs.google.com/java/awt/Robot.html) | This class is used to generate native system input events for the purposes of test automation, self-running demos, and other applications where control of the mouse and keyboard is needed. |
| [**Scrollbar**](http://docs.google.com/java/awt/Scrollbar.html) | The Scrollbar class embodies a scroll bar, a familiar user-interface object. |
| [**ScrollPane**](http://docs.google.com/java/awt/ScrollPane.html) | A container class which implements automatic horizontal and/or vertical scrolling for a single child component. |
| [**ScrollPaneAdjustable**](http://docs.google.com/java/awt/ScrollPaneAdjustable.html) | This class represents the state of a horizontal or vertical scrollbar of a ScrollPane. |
| [**SplashScreen**](http://docs.google.com/java/awt/SplashScreen.html) | The splash screen can be created at application startup, before the Java Virtual Machine (JVM) starts. |
| [**SystemColor**](http://docs.google.com/java/awt/SystemColor.html) | A class to encapsulate symbolic colors representing the color of native GUI objects on a system. |
| [**SystemTray**](http://docs.google.com/java/awt/SystemTray.html) | The SystemTray class represents the system tray for a desktop. |
| [**TextArea**](http://docs.google.com/java/awt/TextArea.html) | A TextArea object is a multi-line region that displays text. |
| [**TextComponent**](http://docs.google.com/java/awt/TextComponent.html) | The TextComponent class is the superclass of any component that allows the editing of some text. |
| [**TextField**](http://docs.google.com/java/awt/TextField.html) | A TextField object is a text component that allows for the editing of a single line of text. |
| [**TexturePaint**](http://docs.google.com/java/awt/TexturePaint.html) | The TexturePaint class provides a way to fill a [Shape](http://docs.google.com/java/awt/Shape.html) with a texture that is specified as a [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html). |
| [**Toolkit**](http://docs.google.com/java/awt/Toolkit.html) | This class is the abstract superclass of all actual implementations of the Abstract Window Toolkit. |
| [**TrayIcon**](http://docs.google.com/java/awt/TrayIcon.html) | A TrayIcon object represents a tray icon that can be added to the [system tray](http://docs.google.com/java/awt/SystemTray.html). |
| [**Window**](http://docs.google.com/java/awt/Window.html) | A Window object is a top-level window with no borders and no menubar. |

| **Enum Summary** | |
| --- | --- |
| [**Component.BaselineResizeBehavior**](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html) | Enumeration of the common ways the baseline of a component can change as the size changes. |
| [**Desktop.Action**](http://docs.google.com/java/awt/Desktop.Action.html) | Represents an action type. |
| [**Dialog.ModalExclusionType**](http://docs.google.com/java/awt/Dialog.ModalExclusionType.html) | Any top-level window can be marked not to be blocked by modal dialogs. |
| [**Dialog.ModalityType**](http://docs.google.com/java/awt/Dialog.ModalityType.html) | Modal dialogs block all input to some top-level windows. |
| [**MultipleGradientPaint.ColorSpaceType**](http://docs.google.com/java/awt/MultipleGradientPaint.ColorSpaceType.html) | The color space in which to perform the gradient interpolation. |
| [**MultipleGradientPaint.CycleMethod**](http://docs.google.com/java/awt/MultipleGradientPaint.CycleMethod.html) | The method to use when painting outside the gradient bounds. |
| [**TrayIcon.MessageType**](http://docs.google.com/java/awt/TrayIcon.MessageType.html) | The message type determines which icon will be displayed in the caption of the message, and a possible system sound a message may generate upon showing. |

| **Exception Summary** | |
| --- | --- |
| [**AWTException**](http://docs.google.com/java/awt/AWTException.html) | Signals that an Absract Window Toolkit exception has occurred. |
| [**FontFormatException**](http://docs.google.com/java/awt/FontFormatException.html) | Thrown by method createFont in the Font class to indicate that the specified font is bad. |
| [**HeadlessException**](http://docs.google.com/java/awt/HeadlessException.html) | Thrown when code that is dependent on a keyboard, display, or mouse is called in an environment that does not support a keyboard, display, or mouse. |
| [**IllegalComponentStateException**](http://docs.google.com/java/awt/IllegalComponentStateException.html) | Signals that an AWT component is not in an appropriate state for the requested operation. |

| **Error Summary** | |
| --- | --- |
| [**AWTError**](http://docs.google.com/java/awt/AWTError.html) | Thrown when a serious Abstract Window Toolkit error has occurred. |

## Package java.awt Description

Contains all of the classes for creating user interfaces and for painting graphics and images. A user interface object such as a button or a scrollbar is called, in AWT terminology, a component. The Component class is the root of all AWT components. See Component for a detailed description of properties that all AWT components share.

Some components fire events when a user interacts with the components. The AWTEvent class and its subclasses are used to represent the events that AWT components can fire. See AWTEvent for a description of the AWT event model.

A container is a component that can contain components and other containers. A con tainer can also have a layout manager that controls the visual placement of components in the container. The AWT package contains several layout manager classes and an interface for building your own layout manager. See Container and LayoutManager for more information.

## Additional Specification

* [The AWT Focus Subsystem](http://docs.google.com/doc-files/FocusSpec.html)
* [The AWT Modality](http://docs.google.com/doc-files/Modality.html)

**Since:** JDK1.0

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV PACKAGE**](http://docs.google.com/java/applet/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/awt/color/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).