| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Checkbox.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 CLASS**](http://docs.google.com/java/awt/CardLayout.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Checkbox.AccessibleAWTCheckbox.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Checkbox.html)    [**NO FRAMES**](http://docs.google.com/Checkbox.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: FIELD | [CONSTR](#26in1rg) | [METHOD](#z337ya) |

## **java.awt**

Class Checkbox

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.Component](http://docs.google.com/java/awt/Component.html)  
 **java.awt.Checkbox**

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

public class **Checkbox**extends [Component](http://docs.google.com/java/awt/Component.html)implements [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

A check box is a graphical component that can be in either an "on" (true) or "off" (false) state. Clicking on a check box changes its state from "on" to "off," or from "off" to "on."

The following code example creates a set of check boxes in a grid layout:

setLayout(new GridLayout(3, 1));  
 add(new Checkbox("one", null, true));  
 add(new Checkbox("two"));  
 add(new Checkbox("three"));

This image depicts the check boxes and grid layout created by this code example:



The button labeled one is in the "on" state, and the other two are in the "off" state. In this example, which uses the GridLayout class, the states of the three check boxes are set independently.

Alternatively, several check boxes can be grouped together under the control of a single object, using the CheckboxGroup class. In a check box group, at most one button can be in the "on" state at any given time. Clicking on a check box to turn it on forces any other check box in the same group that is on into the "off" state.

**Since:** JDK1.0 **See Also:**[GridLayout](http://docs.google.com/java/awt/GridLayout.html), [CheckboxGroup](http://docs.google.com/java/awt/CheckboxGroup.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.Checkbox)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**Checkbox.AccessibleAWTCheckbox**](http://docs.google.com/java/awt/Checkbox.AccessibleAWTCheckbox.html)            This class implements accessibility support for the Checkbox class. |

| **Nested classes/interfaces inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [Component.AccessibleAWTComponent](http://docs.google.com/java/awt/Component.AccessibleAWTComponent.html), [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html), [Component.BltBufferStrategy](http://docs.google.com/java/awt/Component.BltBufferStrategy.html), [Component.FlipBufferStrategy](http://docs.google.com/java/awt/Component.FlipBufferStrategy.html) |

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [BOTTOM\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#BOTTOM_ALIGNMENT), [CENTER\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#CENTER_ALIGNMENT), [LEFT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#LEFT_ALIGNMENT), [RIGHT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#RIGHT_ALIGNMENT), [TOP\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#TOP_ALIGNMENT) |

| **Fields inherited from interface java.awt.image.**[**ImageObserver**](http://docs.google.com/java/awt/image/ImageObserver.html) |
| --- |
| [ABORT](http://docs.google.com/java/awt/image/ImageObserver.html#ABORT), [ALLBITS](http://docs.google.com/java/awt/image/ImageObserver.html#ALLBITS), [ERROR](http://docs.google.com/java/awt/image/ImageObserver.html#ERROR), [FRAMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#FRAMEBITS), [HEIGHT](http://docs.google.com/java/awt/image/ImageObserver.html#HEIGHT), [PROPERTIES](http://docs.google.com/java/awt/image/ImageObserver.html#PROPERTIES), [SOMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#SOMEBITS), [WIDTH](http://docs.google.com/java/awt/image/ImageObserver.html#WIDTH) |

| **Constructor Summary** | |
| --- | --- |
| [**Checkbox**](http://docs.google.com/java/awt/Checkbox.html#Checkbox())()            Creates a check box with an empty string for its label. |
| [**Checkbox**](http://docs.google.com/java/awt/Checkbox.html#Checkbox(java.lang.String))([String](http://docs.google.com/java/lang/String.html) label)            Creates a check box with the specified label. |
| [**Checkbox**](http://docs.google.com/java/awt/Checkbox.html#Checkbox(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) label, boolean state)            Creates a check box with the specified label and sets the specified state. |
| [**Checkbox**](http://docs.google.com/java/awt/Checkbox.html#Checkbox(java.lang.String,%20boolean,%20java.awt.CheckboxGroup))([String](http://docs.google.com/java/lang/String.html) label, boolean state, [CheckboxGroup](http://docs.google.com/java/awt/CheckboxGroup.html) group)            Constructs a Checkbox with the specified label, set to the specified state, and in the specified check box group. |
| [**Checkbox**](http://docs.google.com/java/awt/Checkbox.html#Checkbox(java.lang.String,%20java.awt.CheckboxGroup,%20boolean))([String](http://docs.google.com/java/lang/String.html) label, [CheckboxGroup](http://docs.google.com/java/awt/CheckboxGroup.html) group, boolean state)            Creates a check box with the specified label, in the specified check box group, and set to the specified state. |

| **Method Summary** | |
| --- | --- |
| void | [**addItemListener**](http://docs.google.com/java/awt/Checkbox.html#addItemListener(java.awt.event.ItemListener))([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) l)            Adds the specified item listener to receive item events from this check box. |
| void | [**addNotify**](http://docs.google.com/java/awt/Checkbox.html#addNotify())()            Creates the peer of the Checkbox. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/java/awt/Checkbox.html#getAccessibleContext())()            Gets the AccessibleContext associated with this Checkbox. |
| [CheckboxGroup](http://docs.google.com/java/awt/CheckboxGroup.html) | [**getCheckboxGroup**](http://docs.google.com/java/awt/Checkbox.html#getCheckboxGroup())()            Determines this check box's group. |
| [ItemListener](http://docs.google.com/java/awt/event/ItemListener.html)[] | [**getItemListeners**](http://docs.google.com/java/awt/Checkbox.html#getItemListeners())()            Returns an array of all the item listeners registered on this checkbox. |
| [String](http://docs.google.com/java/lang/String.html) | [**getLabel**](http://docs.google.com/java/awt/Checkbox.html#getLabel())()            Gets the label of this check box. |
| | <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)>  T[] | | --- | | [**getListeners**](http://docs.google.com/java/awt/Checkbox.html#getListeners(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)            Returns an array of all the objects currently registered as *Foo*Listeners upon this Checkbox. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getSelectedObjects**](http://docs.google.com/java/awt/Checkbox.html#getSelectedObjects())()            Returns an array (length 1) containing the checkbox label or null if the checkbox is not selected. |
| boolean | [**getState**](http://docs.google.com/java/awt/Checkbox.html#getState())()            Determines whether this check box is in the "on" or "off" state. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/Checkbox.html#paramString())()            Returns a string representing the state of this Checkbox. |
| protected  void | [**processEvent**](http://docs.google.com/java/awt/Checkbox.html#processEvent(java.awt.AWTEvent))([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)            Processes events on this check box. |
| protected  void | [**processItemEvent**](http://docs.google.com/java/awt/Checkbox.html#processItemEvent(java.awt.event.ItemEvent))([ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html) e)            Processes item events occurring on this check box by dispatching them to any registered ItemListener objects. |
| void | [**removeItemListener**](http://docs.google.com/java/awt/Checkbox.html#removeItemListener(java.awt.event.ItemListener))([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) l)            Removes the specified item listener so that the item listener no longer receives item events from this check box. |
| void | [**setCheckboxGroup**](http://docs.google.com/java/awt/Checkbox.html#setCheckboxGroup(java.awt.CheckboxGroup))([CheckboxGroup](http://docs.google.com/java/awt/CheckboxGroup.html) g)            Sets this check box's group to the specified check box group. |
| void | [**setLabel**](http://docs.google.com/java/awt/Checkbox.html#setLabel(java.lang.String))([String](http://docs.google.com/java/lang/String.html) label)            Sets this check box's label to be the string argument. |
| void | [**setState**](http://docs.google.com/java/awt/Checkbox.html#setState(boolean))(boolean state)            Sets the state of this check box to the specified state. |

| **Methods inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [action](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu)), [addComponentListener](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [addFocusListener](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [addHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [addHierarchyListener](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [addInputMethodListener](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [addKeyListener](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [addMouseListener](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [addMouseMotionListener](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [addMouseWheelListener](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [applyComponentOrientation](http://docs.google.com/java/awt/Component.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Component.html#areFocusTraversalKeysSet(int)), [bounds](http://docs.google.com/java/awt/Component.html#bounds()), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20java.awt.image.ImageObserver)), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [coalesceEvents](http://docs.google.com/java/awt/Component.html#coalesceEvents(java.awt.AWTEvent,%20java.awt.AWTEvent)), [contains](http://docs.google.com/java/awt/Component.html#contains(int,%20int)), [contains](http://docs.google.com/java/awt/Component.html#contains(java.awt.Point)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int,%20java.awt.ImageCapabilities)), [deliverEvent](http://docs.google.com/java/awt/Component.html#deliverEvent(java.awt.Event)), [disable](http://docs.google.com/java/awt/Component.html#disable()), [disableEvents](http://docs.google.com/java/awt/Component.html#disableEvents(long)), [dispatchEvent](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [doLayout](http://docs.google.com/java/awt/Component.html#doLayout()), [enable](http://docs.google.com/java/awt/Component.html#enable()), [enable](http://docs.google.com/java/awt/Component.html#enable(boolean)), [enableEvents](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [enableInputMethods](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20byte,%20byte)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20double,%20double)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20float,%20float)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20int,%20int)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20long,%20long)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20short,%20short)), [getAlignmentX](http://docs.google.com/java/awt/Component.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/java/awt/Component.html#getAlignmentY()), [getBackground](http://docs.google.com/java/awt/Component.html#getBackground()), [getBaseline](http://docs.google.com/java/awt/Component.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/java/awt/Component.html#getBaselineResizeBehavior()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds(java.awt.Rectangle)), [getColorModel](http://docs.google.com/java/awt/Component.html#getColorModel()), [getComponentAt](http://docs.google.com/java/awt/Component.html#getComponentAt(int,%20int)), [getComponentAt](http://docs.google.com/java/awt/Component.html#getComponentAt(java.awt.Point)), [getComponentListeners](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [getComponentOrientation](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [getCursor](http://docs.google.com/java/awt/Component.html#getCursor()), [getDropTarget](http://docs.google.com/java/awt/Component.html#getDropTarget()), [getFocusCycleRootAncestor](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor()), [getFocusListeners](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeys(int)), [getFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled()), [getFont](http://docs.google.com/java/awt/Component.html#getFont()), [getFontMetrics](http://docs.google.com/java/awt/Component.html#getFontMetrics(java.awt.Font)), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphics](http://docs.google.com/java/awt/Component.html#getGraphics()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [getHeight](http://docs.google.com/java/awt/Component.html#getHeight()), [getHierarchyBoundsListeners](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [getHierarchyListeners](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [getIgnoreRepaint](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint()), [getInputContext](http://docs.google.com/java/awt/Component.html#getInputContext()), [getInputMethodListeners](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getInputMethodRequests](http://docs.google.com/java/awt/Component.html#getInputMethodRequests()), [getKeyListeners](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getLocale](http://docs.google.com/java/awt/Component.html#getLocale()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation(java.awt.Point)), [getLocationOnScreen](http://docs.google.com/java/awt/Component.html#getLocationOnScreen()), [getMaximumSize](http://docs.google.com/java/awt/Component.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/java/awt/Component.html#getMinimumSize()), [getMouseListeners](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMousePosition](http://docs.google.com/java/awt/Component.html#getMousePosition()), [getMouseWheelListeners](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getName](http://docs.google.com/java/awt/Component.html#getName()), [getParent](http://docs.google.com/java/awt/Component.html#getParent()), [getPeer](http://docs.google.com/java/awt/Component.html#getPeer()), [getPreferredSize](http://docs.google.com/java/awt/Component.html#getPreferredSize()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [getSize](http://docs.google.com/java/awt/Component.html#getSize()), [getSize](http://docs.google.com/java/awt/Component.html#getSize(java.awt.Dimension)), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [getWidth](http://docs.google.com/java/awt/Component.html#getWidth()), [getX](http://docs.google.com/java/awt/Component.html#getX()), [getY](http://docs.google.com/java/awt/Component.html#getY()), [gotFocus](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object)), [handleEvent](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event)), [hasFocus](http://docs.google.com/java/awt/Component.html#hasFocus()), [hide](http://docs.google.com/java/awt/Component.html#hide()), [imageUpdate](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [inside](http://docs.google.com/java/awt/Component.html#inside(int,%20int)), [invalidate](http://docs.google.com/java/awt/Component.html#invalidate()), [isBackgroundSet](http://docs.google.com/java/awt/Component.html#isBackgroundSet()), [isCursorSet](http://docs.google.com/java/awt/Component.html#isCursorSet()), [isDisplayable](http://docs.google.com/java/awt/Component.html#isDisplayable()), [isDoubleBuffered](http://docs.google.com/java/awt/Component.html#isDoubleBuffered()), [isEnabled](http://docs.google.com/java/awt/Component.html#isEnabled()), [isFocusable](http://docs.google.com/java/awt/Component.html#isFocusable()), [isFocusCycleRoot](http://docs.google.com/java/awt/Component.html#isFocusCycleRoot(java.awt.Container)), [isFocusOwner](http://docs.google.com/java/awt/Component.html#isFocusOwner()), [isFocusTraversable](http://docs.google.com/java/awt/Component.html#isFocusTraversable()), [isFontSet](http://docs.google.com/java/awt/Component.html#isFontSet()), [isForegroundSet](http://docs.google.com/java/awt/Component.html#isForegroundSet()), [isLightweight](http://docs.google.com/java/awt/Component.html#isLightweight()), [isMaximumSizeSet](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet()), [isMinimumSizeSet](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet()), [isOpaque](http://docs.google.com/java/awt/Component.html#isOpaque()), [isPreferredSizeSet](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet()), [isShowing](http://docs.google.com/java/awt/Component.html#isShowing()), [isValid](http://docs.google.com/java/awt/Component.html#isValid()), [isVisible](http://docs.google.com/java/awt/Component.html#isVisible()), [keyDown](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int)), [keyUp](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int)), [layout](http://docs.google.com/java/awt/Component.html#layout()), [list](http://docs.google.com/java/awt/Component.html#list()), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream,%20int)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter,%20int)), [locate](http://docs.google.com/java/awt/Component.html#locate(int,%20int)), [location](http://docs.google.com/java/awt/Component.html#location()), [lostFocus](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object)), [minimumSize](http://docs.google.com/java/awt/Component.html#minimumSize()), [mouseDown](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int)), [mouseDrag](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int)), [mouseEnter](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int)), [mouseExit](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int)), [mouseMove](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int)), [mouseUp](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int)), [move](http://docs.google.com/java/awt/Component.html#move(int,%20int)), [nextFocus](http://docs.google.com/java/awt/Component.html#nextFocus()), [paint](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics)), [paintAll](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics)), [postEvent](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event)), [preferredSize](http://docs.google.com/java/awt/Component.html#preferredSize()), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20java.awt.image.ImageObserver)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [print](http://docs.google.com/java/awt/Component.html#print(java.awt.Graphics)), [printAll](http://docs.google.com/java/awt/Component.html#printAll(java.awt.Graphics)), [processComponentEvent](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [processFocusEvent](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [processHierarchyBoundsEvent](http://docs.google.com/java/awt/Component.html#processHierarchyBoundsEvent(java.awt.event.HierarchyEvent)), [processHierarchyEvent](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [processInputMethodEvent](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [processKeyEvent](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/java/awt/Component.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/java/awt/Component.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [processMouseWheelEvent](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent)), [remove](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent)), [removeComponentListener](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener)), [removeFocusListener](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener)), [removeHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [removeHierarchyListener](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener)), [removeInputMethodListener](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener)), [removeKeyListener](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener)), [removeMouseListener](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener)), [removeMouseMotionListener](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [removeMouseWheelListener](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener)), [removeNotify](http://docs.google.com/java/awt/Component.html#removeNotify()), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [repaint](http://docs.google.com/java/awt/Component.html#repaint()), [repaint](http://docs.google.com/java/awt/Component.html#repaint(int,%20int,%20int,%20int)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long,%20int,%20int,%20int,%20int)), [requestFocus](http://docs.google.com/java/awt/Component.html#requestFocus()), [requestFocus](http://docs.google.com/java/awt/Component.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/java/awt/Component.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/java/awt/Component.html#requestFocusInWindow(boolean)), [reshape](http://docs.google.com/java/awt/Component.html#reshape(int,%20int,%20int,%20int)), [resize](http://docs.google.com/java/awt/Component.html#resize(java.awt.Dimension)), [resize](http://docs.google.com/java/awt/Component.html#resize(int,%20int)), [setBackground](http://docs.google.com/java/awt/Component.html#setBackground(java.awt.Color)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle)), [setComponentOrientation](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)), [setDropTarget](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget)), [setEnabled](http://docs.google.com/java/awt/Component.html#setEnabled(boolean)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeys](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setFont](http://docs.google.com/java/awt/Component.html#setFont(java.awt.Font)), [setForeground](http://docs.google.com/java/awt/Component.html#setForeground(java.awt.Color)), [setIgnoreRepaint](http://docs.google.com/java/awt/Component.html#setIgnoreRepaint(boolean)), [setLocale](http://docs.google.com/java/awt/Component.html#setLocale(java.util.Locale)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [setMaximumSize](http://docs.google.com/java/awt/Component.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/java/awt/Component.html#setMinimumSize(java.awt.Dimension)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setPreferredSize](http://docs.google.com/java/awt/Component.html#setPreferredSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [setVisible](http://docs.google.com/java/awt/Component.html#setVisible(boolean)), [show](http://docs.google.com/java/awt/Component.html#show()), [show](http://docs.google.com/java/awt/Component.html#show(boolean)), [size](http://docs.google.com/java/awt/Component.html#size()), [toString](http://docs.google.com/java/awt/Component.html#toString()), [transferFocus](http://docs.google.com/java/awt/Component.html#transferFocus()), [transferFocusBackward](http://docs.google.com/java/awt/Component.html#transferFocusBackward()), [transferFocusUpCycle](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle()), [update](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics)), [validate](http://docs.google.com/java/awt/Component.html#validate()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### Checkbox

public **Checkbox**()  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates a check box with an empty string for its label. The state of this check box is set to "off," and it is not part of any check box group.

**Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### Checkbox

public **Checkbox**([String](http://docs.google.com/java/lang/String.html) label)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates a check box with the specified label. The state of this check box is set to "off," and it is not part of any check box group.

**Parameters:**label - a string label for this check box, or null for no label. **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### Checkbox

public **Checkbox**([String](http://docs.google.com/java/lang/String.html) label,  
 boolean state)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates a check box with the specified label and sets the specified state. This check box is not part of any check box group.

**Parameters:**label - a string label for this check box, or null for no labelstate - the initial state of this check box **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### Checkbox

public **Checkbox**([String](http://docs.google.com/java/lang/String.html) label,  
 boolean state,  
 [CheckboxGroup](http://docs.google.com/java/awt/CheckboxGroup.html) group)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Constructs a Checkbox with the specified label, set to the specified state, and in the specified check box group.

**Parameters:**label - a string label for this check box, or null for no label.state - the initial state of this check box.group - a check box group for this check box, or null for no group. **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**Since:** JDK1.1 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### Checkbox

public **Checkbox**([String](http://docs.google.com/java/lang/String.html) label,  
 [CheckboxGroup](http://docs.google.com/java/awt/CheckboxGroup.html) group,  
 boolean state)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates a check box with the specified label, in the specified check box group, and set to the specified state.

**Parameters:**label - a string label for this check box, or null for no label.group - a check box group for this check box, or null for no group.state - the initial state of this check box. **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**Since:** JDK1.1 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

| **Method Detail** |
| --- |

### addNotify

public void **addNotify**()

Creates the peer of the Checkbox. The peer allows you to change the look of the Checkbox without changing its functionality.

**Overrides:**[addNotify](http://docs.google.com/java/awt/Component.html#addNotify()) in class [Component](http://docs.google.com/java/awt/Component.html) **See Also:**[Toolkit.createCheckbox(java.awt.Checkbox)](http://docs.google.com/java/awt/Toolkit.html#createCheckbox(java.awt.Checkbox)), [Component.getToolkit()](http://docs.google.com/java/awt/Component.html#getToolkit())

### getLabel

public [String](http://docs.google.com/java/lang/String.html) **getLabel**()

Gets the label of this check box.

**Returns:**the label of this check box, or null if this check box has no label.**See Also:**[setLabel(String)](http://docs.google.com/java/awt/Checkbox.html#setLabel(java.lang.String))

### setLabel

public void **setLabel**([String](http://docs.google.com/java/lang/String.html) label)

Sets this check box's label to be the string argument.

**Parameters:**label - a string to set as the new label, or null for no label.**See Also:**[getLabel()](http://docs.google.com/java/awt/Checkbox.html#getLabel())

### getState

public boolean **getState**()

Determines whether this check box is in the "on" or "off" state. The boolean value true indicates the "on" state, and false indicates the "off" state.

**Returns:**the state of this check box, as a boolean value**See Also:**[setState(boolean)](http://docs.google.com/java/awt/Checkbox.html#setState(boolean))

### setState

public void **setState**(boolean state)

Sets the state of this check box to the specified state. The boolean value true indicates the "on" state, and false indicates the "off" state.

Note that this method should be primarily used to initialize the state of the checkbox. Programmatically setting the state of the checkbox will *not* trigger an ItemEvent. The only way to trigger an ItemEvent is by user interaction.

**Parameters:**state - the boolean state of the check box**See Also:**[getState()](http://docs.google.com/java/awt/Checkbox.html#getState())

### getSelectedObjects

public [Object](http://docs.google.com/java/lang/Object.html)[] **getSelectedObjects**()

Returns an array (length 1) containing the checkbox label or null if the checkbox is not selected.

**Specified by:**[getSelectedObjects](http://docs.google.com/java/awt/ItemSelectable.html#getSelectedObjects()) in interface [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html) **See Also:**[ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html)

### getCheckboxGroup

public [CheckboxGroup](http://docs.google.com/java/awt/CheckboxGroup.html) **getCheckboxGroup**()

Determines this check box's group.

**Returns:**this check box's group, or null if the check box is not part of a check box group.**See Also:**[setCheckboxGroup(CheckboxGroup)](http://docs.google.com/java/awt/Checkbox.html#setCheckboxGroup(java.awt.CheckboxGroup))

### setCheckboxGroup

public void **setCheckboxGroup**([CheckboxGroup](http://docs.google.com/java/awt/CheckboxGroup.html) g)

Sets this check box's group to the specified check box group. If this check box is already in a different check box group, it is first taken out of that group.

If the state of this check box is true and the new group already has a check box selected, this check box's state is changed to false. If the state of this check box is true and the new group has no check box selected, this check box becomes the selected checkbox for the new group and its state is true.

**Parameters:**g - the new check box group, or null to remove this check box from any check box group**See Also:**[getCheckboxGroup()](http://docs.google.com/java/awt/Checkbox.html#getCheckboxGroup())

### addItemListener

public void **addItemListener**([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) l)

Adds the specified item listener to receive item events from this check box. Item events are sent to listeners in response to user input, but not in response to calls to setState(). If l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Specified by:**[addItemListener](http://docs.google.com/java/awt/ItemSelectable.html#addItemListener(java.awt.event.ItemListener)) in interface [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html) **Parameters:**l - the item listener**Since:** JDK1.1 **See Also:**[removeItemListener(java.awt.event.ItemListener)](http://docs.google.com/java/awt/Checkbox.html#removeItemListener(java.awt.event.ItemListener)), [getItemListeners()](http://docs.google.com/java/awt/Checkbox.html#getItemListeners()), [setState(boolean)](http://docs.google.com/java/awt/Checkbox.html#setState(boolean)), [ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html), [ItemListener](http://docs.google.com/java/awt/event/ItemListener.html)

### removeItemListener

public void **removeItemListener**([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) l)

Removes the specified item listener so that the item listener no longer receives item events from this check box. If l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Specified by:**[removeItemListener](http://docs.google.com/java/awt/ItemSelectable.html#removeItemListener(java.awt.event.ItemListener)) in interface [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html) **Parameters:**l - the item listener**Since:** JDK1.1 **See Also:**[addItemListener(java.awt.event.ItemListener)](http://docs.google.com/java/awt/Checkbox.html#addItemListener(java.awt.event.ItemListener)), [getItemListeners()](http://docs.google.com/java/awt/Checkbox.html#getItemListeners()), [ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html), [ItemListener](http://docs.google.com/java/awt/event/ItemListener.html)

### getItemListeners

public [ItemListener](http://docs.google.com/java/awt/event/ItemListener.html)[] **getItemListeners**()

Returns an array of all the item listeners registered on this checkbox.

**Returns:**all of this checkbox's ItemListeners or an empty array if no item listeners are currently registered**Since:** 1.4 **See Also:**[addItemListener(java.awt.event.ItemListener)](http://docs.google.com/java/awt/Checkbox.html#addItemListener(java.awt.event.ItemListener)), [removeItemListener(java.awt.event.ItemListener)](http://docs.google.com/java/awt/Checkbox.html#removeItemListener(java.awt.event.ItemListener)), [ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html), [ItemListener](http://docs.google.com/java/awt/event/ItemListener.html)

### getListeners

public <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)> T[] **getListeners**([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)

Returns an array of all the objects currently registered as *Foo*Listeners upon this Checkbox. *Foo*Listeners are registered using the add*Foo*Listener method.

You can specify the listenerType argument with a class literal, such as *Foo*Listener.class. For example, you can query a Checkbox c for its item listeners with the following code:

ItemListener[] ils = (ItemListener[])(c.getListeners(ItemListener.class));

If no such listeners exist, this method returns an empty array.

**Overrides:**[getListeners](http://docs.google.com/java/awt/Component.html#getListeners(java.lang.Class)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**listenerType - the type of listeners requested; this parameter should specify an interface that descends from java.util.EventListener **Returns:**an array of all objects registered as *Foo*Listeners on this checkbox, or an empty array if no such listeners have been added **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if listenerType doesn't specify a class or interface that implements java.util.EventListener**Since:** 1.3 **See Also:**[getItemListeners()](http://docs.google.com/java/awt/Checkbox.html#getItemListeners())

### processEvent

protected void **processEvent**([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)

Processes events on this check box. If the event is an instance of ItemEvent, this method invokes the processItemEvent method. Otherwise, it calls its superclass's processEvent method.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Overrides:**[processEvent](http://docs.google.com/java/awt/Component.html#processEvent(java.awt.AWTEvent)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**e - the event**Since:** JDK1.1 **See Also:**[ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html), [processItemEvent(java.awt.event.ItemEvent)](http://docs.google.com/java/awt/Checkbox.html#processItemEvent(java.awt.event.ItemEvent))

### processItemEvent

protected void **processItemEvent**([ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html) e)

Processes item events occurring on this check box by dispatching them to any registered ItemListener objects.

This method is not called unless item events are enabled for this component. Item events are enabled when one of the following occurs:

* An ItemListener object is registered via addItemListener.
* Item events are enabled via enableEvents.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the item event**Since:** JDK1.1 **See Also:**[ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html), [ItemListener](http://docs.google.com/java/awt/event/ItemListener.html), [addItemListener(java.awt.event.ItemListener)](http://docs.google.com/java/awt/Checkbox.html#addItemListener(java.awt.event.ItemListener)), [Component.enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### paramString

protected [String](http://docs.google.com/java/lang/String.html) **paramString**()

Returns a string representing the state of this Checkbox. This method is intended to be used only for debugging purposes, and the content and format of the returned string may vary between implementations. The returned string may be empty but may not be null.

**Overrides:**[paramString](http://docs.google.com/java/awt/Component.html#paramString()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**the parameter string of this check box

### getAccessibleContext

public [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) **getAccessibleContext**()

Gets the AccessibleContext associated with this Checkbox. For checkboxes, the AccessibleContext takes the form of an AccessibleAWTCheckbox. A new AccessibleAWTCheckbox is created if necessary.

**Specified by:**[getAccessibleContext](http://docs.google.com/javax/accessibility/Accessible.html#getAccessibleContext()) in interface [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)**Overrides:**[getAccessibleContext](http://docs.google.com/java/awt/Component.html#getAccessibleContext()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**an AccessibleAWTCheckbox that serves as the AccessibleContext of this Checkbox**Since:** 1.3

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Checkbox.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 CLASS**](http://docs.google.com/java/awt/CardLayout.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Checkbox.AccessibleAWTCheckbox.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Checkbox.html)    [**NO FRAMES**](http://docs.google.com/Checkbox.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: FIELD | [CONSTR](#26in1rg) | [METHOD](#z337ya) |

[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).