| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Container.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/CompositeContext.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Container.AccessibleAWTContainer.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Container.html)    [**NO FRAMES**](http://docs.google.com/Container.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: FIELD | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

## **java.awt**

Class Container

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

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [Serializable](http://docs.google.com/java/io/Serializable.html) **Direct Known Subclasses:** [BasicSplitPaneDivider](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html), [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html), [DefaultTreeCellEditor.EditorContainer](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.EditorContainer.html), [JComponent](http://docs.google.com/javax/swing/JComponent.html), [Panel](http://docs.google.com/java/awt/Panel.html), [ScrollPane](http://docs.google.com/java/awt/ScrollPane.html), [Window](http://docs.google.com/java/awt/Window.html)

public class **Container**extends [Component](http://docs.google.com/java/awt/Component.html)

A generic Abstract Window Toolkit(AWT) container object is a component that can contain other AWT components.

Components added to a container are tracked in a list. The order of the list will define the components' front-to-back stacking order within the container. If no index is specified when adding a component to a container, it will be added to the end of the list (and hence to the bottom of the stacking order).

**Note**: For details on the focus subsystem, see  [How to Use the Focus Subsystem](http://java.sun.com/docs/books/tutorial/uiswing/misc/focus.html), a section in *The Java Tutorial*, and the [Focus Specification](http://docs.google.com/java/awt/doc-files/FocusSpec.html) for more information.

**Since:** JDK1.0 **See Also:**[add(java.awt.Component, int)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [getComponent(int)](http://docs.google.com/java/awt/Container.html#getComponent(int)), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.Container)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**Container.AccessibleAWTContainer**](http://docs.google.com/java/awt/Container.AccessibleAWTContainer.html)            Inner class of Container used to provide default support for accessibility. |

| **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** | |
| --- | --- |
| [**Container**](http://docs.google.com/java/awt/Container.html#Container())()            Constructs a new Container. |

| **Method Summary** | |
| --- | --- |
| [Component](http://docs.google.com/java/awt/Component.html) | [**add**](http://docs.google.com/java/awt/Container.html#add(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            Appends the specified component to the end of this container. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**add**](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int))([Component](http://docs.google.com/java/awt/Component.html) comp, int index)            Adds the specified component to this container at the given position. |
| void | [**add**](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) comp, [Object](http://docs.google.com/java/lang/Object.html) constraints)            Adds the specified component to the end of this container. |
| void | [**add**](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object,%20int))([Component](http://docs.google.com/java/awt/Component.html) comp, [Object](http://docs.google.com/java/lang/Object.html) constraints, int index)            Adds the specified component to this container with the specified constraints at the specified index. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**add**](http://docs.google.com/java/awt/Container.html#add(java.lang.String,%20java.awt.Component))([String](http://docs.google.com/java/lang/String.html) name, [Component](http://docs.google.com/java/awt/Component.html) comp)            Adds the specified component to this container. |
| void | [**addContainerListener**](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener))([ContainerListener](http://docs.google.com/java/awt/event/ContainerListener.html) l)            Adds the specified container listener to receive container events from this container. |
| protected  void | [**addImpl**](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int))([Component](http://docs.google.com/java/awt/Component.html) comp, [Object](http://docs.google.com/java/lang/Object.html) constraints, int index)            Adds the specified component to this container at the specified index. |
| void | [**addNotify**](http://docs.google.com/java/awt/Container.html#addNotify())()            Makes this Container displayable by connecting it to a native screen resource. |
| void | [**addPropertyChangeListener**](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.beans.PropertyChangeListener))([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Adds a PropertyChangeListener to the listener list. |
| void | [**addPropertyChangeListener**](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener))([String](http://docs.google.com/java/lang/String.html) propertyName, [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Adds a PropertyChangeListener to the listener list for a specific property. |
| void | [**applyComponentOrientation**](http://docs.google.com/java/awt/Container.html#applyComponentOrientation(java.awt.ComponentOrientation))([ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html) o)            Sets the ComponentOrientation property of this container and all components contained within it. |
| boolean | [**areFocusTraversalKeysSet**](http://docs.google.com/java/awt/Container.html#areFocusTraversalKeysSet(int))(int id)            Returns whether the Set of focus traversal keys for the given focus traversal operation has been explicitly defined for this Container. |
| int | [**countComponents**](http://docs.google.com/java/awt/Container.html#countComponents())()  **Deprecated.** *As of JDK version 1.1, replaced by getComponentCount().* |
| void | [**deliverEvent**](http://docs.google.com/java/awt/Container.html#deliverEvent(java.awt.Event))([Event](http://docs.google.com/java/awt/Event.html) e)  **Deprecated.** *As of JDK version 1.1, replaced by dispatchEvent(AWTEvent e)* |
| void | [**doLayout**](http://docs.google.com/java/awt/Container.html#doLayout())()            Causes this container to lay out its components. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**findComponentAt**](http://docs.google.com/java/awt/Container.html#findComponentAt(int,%20int))(int x, int y)            Locates the visible child component that contains the specified position. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**findComponentAt**](http://docs.google.com/java/awt/Container.html#findComponentAt(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Locates the visible child component that contains the specified point. |
| float | [**getAlignmentX**](http://docs.google.com/java/awt/Container.html#getAlignmentX())()            Returns the alignment along the x axis. |
| float | [**getAlignmentY**](http://docs.google.com/java/awt/Container.html#getAlignmentY())()            Returns the alignment along the y axis. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponent**](http://docs.google.com/java/awt/Container.html#getComponent(int))(int n)            Gets the nth component in this container. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponentAt**](http://docs.google.com/java/awt/Container.html#getComponentAt(int,%20int))(int x, int y)            Locates the component that contains the x,y position. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponentAt**](http://docs.google.com/java/awt/Container.html#getComponentAt(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Gets the component that contains the specified point. |
| int | [**getComponentCount**](http://docs.google.com/java/awt/Container.html#getComponentCount())()            Gets the number of components in this panel. |
| [Component](http://docs.google.com/java/awt/Component.html)[] | [**getComponents**](http://docs.google.com/java/awt/Container.html#getComponents())()            Gets all the components in this container. |
| int | [**getComponentZOrder**](http://docs.google.com/java/awt/Container.html#getComponentZOrder(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            Returns the z-order index of the component inside the container. |
| [ContainerListener](http://docs.google.com/java/awt/event/ContainerListener.html)[] | [**getContainerListeners**](http://docs.google.com/java/awt/Container.html#getContainerListeners())()            Returns an array of all the container listeners registered on this container. |
| [Set](http://docs.google.com/java/util/Set.html)<[AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html)> | [**getFocusTraversalKeys**](http://docs.google.com/java/awt/Container.html#getFocusTraversalKeys(int))(int id)            Returns the Set of focus traversal keys for a given traversal operation for this Container. |
| [FocusTraversalPolicy](http://docs.google.com/java/awt/FocusTraversalPolicy.html) | [**getFocusTraversalPolicy**](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy())()            Returns the focus traversal policy that will manage keyboard traversal of this Container's children, or null if this Container is not a focus cycle root. |
| [Insets](http://docs.google.com/java/awt/Insets.html) | [**getInsets**](http://docs.google.com/java/awt/Container.html#getInsets())()            Determines the insets of this container, which indicate the size of the container's border. |
| [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) | [**getLayout**](http://docs.google.com/java/awt/Container.html#getLayout())()            Gets the layout manager for this container. |
| | <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)>  T[] | | --- | | [**getListeners**](http://docs.google.com/java/awt/Container.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 Container. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/java/awt/Container.html#getMaximumSize())()            Returns the maximum size of this container. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/java/awt/Container.html#getMinimumSize())()            Returns the minimum size of this container. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getMousePosition**](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean))(boolean allowChildren)            Returns the position of the mouse pointer in this Container's coordinate space if the Container is under the mouse pointer, otherwise returns null. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/java/awt/Container.html#getPreferredSize())()            Returns the preferred size of this container. |
| [Insets](http://docs.google.com/java/awt/Insets.html) | [**insets**](http://docs.google.com/java/awt/Container.html#insets())()  **Deprecated.** *As of JDK version 1.1, replaced by getInsets().* |
| void | [**invalidate**](http://docs.google.com/java/awt/Container.html#invalidate())()            Invalidates the container. |
| boolean | [**isAncestorOf**](http://docs.google.com/java/awt/Container.html#isAncestorOf(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Checks if the component is contained in the component hierarchy of this container. |
| boolean | [**isFocusCycleRoot**](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot())()            Returns whether this Container is the root of a focus traversal cycle. |
| boolean | [**isFocusCycleRoot**](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) container)            Returns whether the specified Container is the focus cycle root of this Container's focus traversal cycle. |
| boolean | [**isFocusTraversalPolicyProvider**](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicyProvider())()            Returns whether this container provides focus traversal policy. |
| boolean | [**isFocusTraversalPolicySet**](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicySet())()            Returns whether the focus traversal policy has been explicitly set for this Container. |
| void | [**layout**](http://docs.google.com/java/awt/Container.html#layout())()  **Deprecated.** *As of JDK version 1.1, replaced by doLayout().* |
| void | [**list**](http://docs.google.com/java/awt/Container.html#list(java.io.PrintStream,%20int))([PrintStream](http://docs.google.com/java/io/PrintStream.html) out, int indent)            Prints a listing of this container to the specified output stream. |
| void | [**list**](http://docs.google.com/java/awt/Container.html#list(java.io.PrintWriter,%20int))([PrintWriter](http://docs.google.com/java/io/PrintWriter.html) out, int indent)            Prints out a list, starting at the specified indentation, to the specified print writer. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**locate**](http://docs.google.com/java/awt/Container.html#locate(int,%20int))(int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by getComponentAt(int, int).* |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumSize**](http://docs.google.com/java/awt/Container.html#minimumSize())()  **Deprecated.** *As of JDK version 1.1, replaced by getMinimumSize().* |
| void | [**paint**](http://docs.google.com/java/awt/Container.html#paint(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Paints the container. |
| void | [**paintComponents**](http://docs.google.com/java/awt/Container.html#paintComponents(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Paints each of the components in this container. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/Container.html#paramString())()            Returns a string representing the state of this Container. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredSize**](http://docs.google.com/java/awt/Container.html#preferredSize())()  **Deprecated.** *As of JDK version 1.1, replaced by getPreferredSize().* |
| void | [**print**](http://docs.google.com/java/awt/Container.html#print(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Prints the container. |
| void | [**printComponents**](http://docs.google.com/java/awt/Container.html#printComponents(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Prints each of the components in this container. |
| protected  void | [**processContainerEvent**](http://docs.google.com/java/awt/Container.html#processContainerEvent(java.awt.event.ContainerEvent))([ContainerEvent](http://docs.google.com/java/awt/event/ContainerEvent.html) e)            Processes container events occurring on this container by dispatching them to any registered ContainerListener objects. |
| protected  void | [**processEvent**](http://docs.google.com/java/awt/Container.html#processEvent(java.awt.AWTEvent))([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)            Processes events on this container. |
| void | [**remove**](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            Removes the specified component from this container. |
| void | [**remove**](http://docs.google.com/java/awt/Container.html#remove(int))(int index)            Removes the component, specified by index, from this container. |
| void | [**removeAll**](http://docs.google.com/java/awt/Container.html#removeAll())()            Removes all the components from this container. |
| void | [**removeContainerListener**](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener))([ContainerListener](http://docs.google.com/java/awt/event/ContainerListener.html) l)            Removes the specified container listener so it no longer receives container events from this container. |
| void | [**removeNotify**](http://docs.google.com/java/awt/Container.html#removeNotify())()            Makes this Container undisplayable by removing its connection to its native screen resource. |
| void | [**setComponentZOrder**](http://docs.google.com/java/awt/Container.html#setComponentZOrder(java.awt.Component,%20int))([Component](http://docs.google.com/java/awt/Component.html) comp, int index)            Moves the specified component to the specified z-order index in the container. |
| void | [**setFocusCycleRoot**](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean))(boolean focusCycleRoot)            Sets whether this Container is the root of a focus traversal cycle. |
| void | [**setFocusTraversalKeys**](http://docs.google.com/java/awt/Container.html#setFocusTraversalKeys(int,%20java.util.Set))(int id, [Set](http://docs.google.com/java/util/Set.html)<? extends [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html)> keystrokes)            Sets the focus traversal keys for a given traversal operation for this Container. |
| void | [**setFocusTraversalPolicy**](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy))([FocusTraversalPolicy](http://docs.google.com/java/awt/FocusTraversalPolicy.html) policy)            Sets the focus traversal policy that will manage keyboard traversal of this Container's children, if this Container is a focus cycle root. |
| void | [**setFocusTraversalPolicyProvider**](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean))(boolean provider)            Sets whether this container will be used to provide focus traversal policy. |
| void | [**setFont**](http://docs.google.com/java/awt/Container.html#setFont(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) f)            Sets the font of this container. |
| void | [**setLayout**](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager))([LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) mgr)            Sets the layout manager for this container. |
| void | [**transferFocusBackward**](http://docs.google.com/java/awt/Container.html#transferFocusBackward())()            Transfers the focus to the previous component, as though this Component were the focus owner. |
| void | [**transferFocusDownCycle**](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle())()            Transfers the focus down one focus traversal cycle. |
| void | [**update**](http://docs.google.com/java/awt/Container.html#update(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Updates the container. |
| void | [**validate**](http://docs.google.com/java/awt/Container.html#validate())()            Validates this container and all of its subcomponents. |
| protected  void | [**validateTree**](http://docs.google.com/java/awt/Container.html#validateTree())()            Recursively descends the container tree and recomputes the layout for any subtrees marked as needing it (those marked as invalid). |

| **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)), [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)), [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)), [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)), [getAccessibleContext](http://docs.google.com/java/awt/Component.html#getAccessibleContext()), [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()), [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()), [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()), [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()), [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)), [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()), [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)), [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.PrintWriter)), [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)), [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()), [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)), [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)), [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)), [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)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [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()), [transferFocusUpCycle](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle()) |

| **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** |
| --- |

### Container

public **Container**()

Constructs a new Container. Containers can be extended directly, but are lightweight in this case and must be contained by a parent somewhere higher up in the component tree that is native. (such as Frame for example).

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

### getComponentCount

public int **getComponentCount**()

Gets the number of components in this panel.

**Returns:**the number of components in this panel.**Since:** JDK1.1 **See Also:**[getComponent(int)](http://docs.google.com/java/awt/Container.html#getComponent(int))

### countComponents

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public int **countComponents**()

**Deprecated.** *As of JDK version 1.1, replaced by getComponentCount().*

### getComponent

public [Component](http://docs.google.com/java/awt/Component.html) **getComponent**(int n)

Gets the nth component in this container.

**Parameters:**n - the index of the component to get. **Returns:**the nth component in this container. **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the nth value does not exist.

### getComponents

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

Gets all the components in this container.

**Returns:**an array of all the components in this container.

### getInsets

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

Determines the insets of this container, which indicate the size of the container's border.

A Frame object, for example, has a top inset that corresponds to the height of the frame's title bar.

**Returns:**the insets of this container.**Since:** JDK1.1 **See Also:**[Insets](http://docs.google.com/java/awt/Insets.html), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### insets

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Insets](http://docs.google.com/java/awt/Insets.html) **insets**()

**Deprecated.** *As of JDK version 1.1, replaced by getInsets().*

### add

public [Component](http://docs.google.com/java/awt/Component.html) **add**([Component](http://docs.google.com/java/awt/Component.html) comp)

Appends the specified component to the end of this container. This is a convenience method for [addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)).

Note: If a component has been added to a container that has been displayed, validate must be called on that container to display the new component. If multiple components are being added, you can improve efficiency by calling validate only once, after all the components have been added.

**Parameters:**comp - the component to be added **Returns:**the component argument **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if comp is null**See Also:**[addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [validate()](http://docs.google.com/java/awt/Container.html#validate()), [JComponent.revalidate()](http://docs.google.com/javax/swing/JComponent.html#revalidate())

### add

public [Component](http://docs.google.com/java/awt/Component.html) **add**([String](http://docs.google.com/java/lang/String.html) name,  
 [Component](http://docs.google.com/java/awt/Component.html) comp)

Adds the specified component to this container. This is a convenience method for [addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)).

This method is obsolete as of 1.1. Please use the method add(Component, Object) instead.

**Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if comp is null**See Also:**[add(Component, Object)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object))

### add

public [Component](http://docs.google.com/java/awt/Component.html) **add**([Component](http://docs.google.com/java/awt/Component.html) comp,  
 int index)

Adds the specified component to this container at the given position. This is a convenience method for [addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)).

Note: If a component has been added to a container that has been displayed, validate must be called on that container to display the new component. If multiple components are being added, you can improve efficiency by calling validate only once, after all the components have been added.

**Parameters:**comp - the component to be addedindex - the position at which to insert the component, or -1 to append the component to the end **Returns:**the component comp **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if comp is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if index is invalid (see [addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)) for details)**See Also:**[addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [remove(int)](http://docs.google.com/java/awt/Container.html#remove(int)), [validate()](http://docs.google.com/java/awt/Container.html#validate()), [JComponent.revalidate()](http://docs.google.com/javax/swing/JComponent.html#revalidate())

### setComponentZOrder

public void **setComponentZOrder**([Component](http://docs.google.com/java/awt/Component.html) comp,  
 int index)

Moves the specified component to the specified z-order index in the container. The z-order determines the order that components are painted; the component with the highest z-order paints first and the component with the lowest z-order paints last. Where components overlap, the component with the lower z-order paints over the component with the higher z-order.

If the component is a child of some other container, it is removed from that container before being added to this container. The important difference between this method and java.awt.Container.add(Component, int) is that this method doesn't call removeNotify on the component while removing it from its previous container unless necessary and when allowed by the underlying native windowing system. This way, if the component has the keyboard focus, it maintains the focus when moved to the new position.

This property is guaranteed to apply only to lightweight non-Container components.

**Note**: Not all platforms support changing the z-order of heavyweight components from one container into another without the call to removeNotify. There is no way to detect whether a platform supports this, so developers shouldn't make any assumptions.

**Parameters:**comp - the component to be movedindex - the position in the container's list to insert the component, where getComponentCount() appends to the end **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if comp is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if comp is one of the container's parents [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if index is not in the range [0, getComponentCount()] for moving between containers, or not in the range [0, getComponentCount()-1] for moving inside a container [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if adding a container to itself [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if adding a Window to a container**Since:** 1.5 **See Also:**[getComponentZOrder(java.awt.Component)](http://docs.google.com/java/awt/Container.html#getComponentZOrder(java.awt.Component))

### getComponentZOrder

public int **getComponentZOrder**([Component](http://docs.google.com/java/awt/Component.html) comp)

Returns the z-order index of the component inside the container. The higher a component is in the z-order hierarchy, the lower its index. The component with the lowest z-order index is painted last, above all other child components.

**Parameters:**comp - the component being queried **Returns:**the z-order index of the component; otherwise returns -1 if the component is null or doesn't belong to the container**Since:** 1.5 **See Also:**[setComponentZOrder(java.awt.Component, int)](http://docs.google.com/java/awt/Container.html#setComponentZOrder(java.awt.Component,%20int))

### add

public void **add**([Component](http://docs.google.com/java/awt/Component.html) comp,  
 [Object](http://docs.google.com/java/lang/Object.html) constraints)

Adds the specified component to the end of this container. Also notifies the layout manager to add the component to this container's layout using the specified constraints object. This is a convenience method for [addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)).

Note: If a component has been added to a container that has been displayed, validate must be called on that container to display the new component. If multiple components are being added, you can improve efficiency by calling validate only once, after all the components have been added.

**Parameters:**comp - the component to be addedconstraints - an object expressing layout contraints for this component **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if comp is null**Since:** JDK1.1 **See Also:**[addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [validate()](http://docs.google.com/java/awt/Container.html#validate()), [JComponent.revalidate()](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### add

public void **add**([Component](http://docs.google.com/java/awt/Component.html) comp,  
 [Object](http://docs.google.com/java/lang/Object.html) constraints,  
 int index)

Adds the specified component to this container with the specified constraints at the specified index. Also notifies the layout manager to add the component to the this container's layout using the specified constraints object. This is a convenience method for [addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)).

Note: If a component has been added to a container that has been displayed, validate must be called on that container to display the new component. If multiple components are being added, you can improve efficiency by calling validate only once, after all the components have been added.

**Parameters:**comp - the component to be addedconstraints - an object expressing layout contraints for thisindex - the position in the container's list at which to insert the component; -1 means insert at the end component **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if comp is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if index is invalid (see [addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)) for details)**See Also:**[addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [validate()](http://docs.google.com/java/awt/Container.html#validate()), [JComponent.revalidate()](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [remove(int)](http://docs.google.com/java/awt/Container.html#remove(int)), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### addImpl

protected void **addImpl**([Component](http://docs.google.com/java/awt/Component.html) comp,  
 [Object](http://docs.google.com/java/lang/Object.html) constraints,  
 int index)

Adds the specified component to this container at the specified index. This method also notifies the layout manager to add the component to this container's layout using the specified constraints object via the addLayoutComponent method.

The constraints are defined by the particular layout manager being used. For example, the BorderLayout class defines five constraints: BorderLayout.NORTH, BorderLayout.SOUTH, BorderLayout.EAST, BorderLayout.WEST, and BorderLayout.CENTER.

The GridBagLayout class requires a GridBagConstraints object. Failure to pass the correct type of constraints object results in an IllegalArgumentException.

If the current layout manager implements LayoutManager2, then [LayoutManager2.addLayoutComponent(Component,Object)](http://docs.google.com/java/awt/LayoutManager2.html#addLayoutComponent(java.awt.Component,%20java.lang.Object)) is invoked on it. If the current layout manager does not implement LayoutManager2, and constraints is a String, then [LayoutManager.addLayoutComponent(String,Component)](http://docs.google.com/java/awt/LayoutManager.html#addLayoutComponent(java.lang.String,%20java.awt.Component)) is invoked on it.

If the component is not an ancestor of this container and has a non-null parent, it is removed from its current parent before it is added to this container.

This is the method to override if a program needs to track every add request to a container as all other add methods defer to this one. An overriding method should usually include a call to the superclass's version of the method:

super.addImpl(comp, constraints, index)

**Parameters:**comp - the component to be addedconstraints - an object expressing layout constraints for this componentindex - the position in the container's list at which to insert the component, where -1 means append to the end **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if index is invalid; if comp is a child of this container, the valid range is [-1, getComponentCount()-1]; if component is not a child of this container, the valid range is [-1, getComponentCount()] [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if comp is an ancestor of this container [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if adding a window to a container [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if comp is null**Since:** JDK1.1 **See Also:**[add(Component)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [add(Component, int)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [add(Component, java.lang.Object)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html)

### remove

public void **remove**(int index)

Removes the component, specified by index, from this container. This method also notifies the layout manager to remove the component from this container's layout via the removeLayoutComponent method.

Note: If a component has been removed from a container that had been displayed, [validate()](http://docs.google.com/java/awt/Container.html#validate()) must be called on that container to reflect changes. If multiple components are being removed, you can improve efficiency by calling [validate()](http://docs.google.com/java/awt/Container.html#validate()) only once, after all the components have been removed.

**Parameters:**index - the index of the component to be removed **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if index is not in range [0, getComponentCount()-1]**Since:** JDK1.1 **See Also:**[add(java.awt.Component)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [validate()](http://docs.google.com/java/awt/Container.html#validate()), [getComponentCount()](http://docs.google.com/java/awt/Container.html#getComponentCount())

### remove

public void **remove**([Component](http://docs.google.com/java/awt/Component.html) comp)

Removes the specified component from this container. This method also notifies the layout manager to remove the component from this container's layout via the removeLayoutComponent method.

Note: If a component has been removed from a container that had been displayed, [validate()](http://docs.google.com/java/awt/Container.html#validate()) must be called on that container to reflect changes. If multiple components are being removed, you can improve efficiency by calling [validate()](http://docs.google.com/java/awt/Container.html#validate()) only once, after all the components have been removed.

**Parameters:**comp - the component to be removed**See Also:**[add(java.awt.Component)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [validate()](http://docs.google.com/java/awt/Container.html#validate()), [remove(int)](http://docs.google.com/java/awt/Container.html#remove(int))

### removeAll

public void **removeAll**()

Removes all the components from this container. This method also notifies the layout manager to remove the components from this container's layout via the removeLayoutComponent method.

**See Also:**[add(java.awt.Component)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [remove(int)](http://docs.google.com/java/awt/Container.html#remove(int))

### getLayout

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

Gets the layout manager for this container.

**See Also:**[doLayout()](http://docs.google.com/java/awt/Container.html#doLayout()), [setLayout(java.awt.LayoutManager)](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager))

### setLayout

public void **setLayout**([LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) mgr)

Sets the layout manager for this container.

**Parameters:**mgr - the specified layout manager**See Also:**[doLayout()](http://docs.google.com/java/awt/Container.html#doLayout()), [getLayout()](http://docs.google.com/java/awt/Container.html#getLayout())

### doLayout

public void **doLayout**()

Causes this container to lay out its components. Most programs should not call this method directly, but should invoke the validate method instead.

**Overrides:**[doLayout](http://docs.google.com/java/awt/Component.html#doLayout()) in class [Component](http://docs.google.com/java/awt/Component.html) **Since:** JDK1.1 **See Also:**[LayoutManager.layoutContainer(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#layoutContainer(java.awt.Container)), [setLayout(java.awt.LayoutManager)](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)), [validate()](http://docs.google.com/java/awt/Container.html#validate())

### layout

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **layout**()

**Deprecated.** *As of JDK version 1.1, replaced by doLayout().*

**Overrides:**[layout](http://docs.google.com/java/awt/Component.html#layout()) in class [Component](http://docs.google.com/java/awt/Component.html)

### invalidate

public void **invalidate**()

Invalidates the container. The container and all parents above it are marked as needing to be laid out. This method can be called often, so it needs to execute quickly.

If the LayoutManager installed on this container is an instance of LayoutManager2, then [LayoutManager2.invalidateLayout(Container)](http://docs.google.com/java/awt/LayoutManager2.html#invalidateLayout(java.awt.Container)) is invoked on it supplying this Container as the argument.

**Overrides:**[invalidate](http://docs.google.com/java/awt/Component.html#invalidate()) in class [Component](http://docs.google.com/java/awt/Component.html) **See Also:**[validate()](http://docs.google.com/java/awt/Container.html#validate()), [layout()](http://docs.google.com/java/awt/Container.html#layout()), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [LayoutManager2.invalidateLayout(Container)](http://docs.google.com/java/awt/LayoutManager2.html#invalidateLayout(java.awt.Container))

### validate

public void **validate**()

Validates this container and all of its subcomponents.

The validate method is used to cause a container to lay out its subcomponents again. It should be invoked when this container's subcomponents are modified (added to or removed from the container, or layout-related information changed) after the container has been displayed.

If this Container is not valid, this method invokes the validateTree method and marks this Container as valid. Otherwise, no action is performed.

**Overrides:**[validate](http://docs.google.com/java/awt/Component.html#validate()) in class [Component](http://docs.google.com/java/awt/Component.html) **See Also:**[add(java.awt.Component)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [Component.invalidate()](http://docs.google.com/java/awt/Component.html#invalidate()), [JComponent.revalidate()](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [validateTree()](http://docs.google.com/java/awt/Container.html#validateTree())

### validateTree

protected void **validateTree**()

Recursively descends the container tree and recomputes the layout for any subtrees marked as needing it (those marked as invalid). Synchronization should be provided by the method that calls this one: validate.

**See Also:**[doLayout()](http://docs.google.com/java/awt/Container.html#doLayout()), [validate()](http://docs.google.com/java/awt/Container.html#validate())

### setFont

public void **setFont**([Font](http://docs.google.com/java/awt/Font.html) f)

Sets the font of this container.

**Overrides:**[setFont](http://docs.google.com/java/awt/Component.html#setFont(java.awt.Font)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**f - The font to become this container's font.**Since:** JDK1.0 **See Also:**[Component.getFont()](http://docs.google.com/java/awt/Component.html#getFont())

### getPreferredSize

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

Returns the preferred size of this container. If the preferred size has not been set explicitly by [Component.setPreferredSize(Dimension)](http://docs.google.com/java/awt/Component.html#setPreferredSize(java.awt.Dimension)) and this Container has a non-null [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), then [LayoutManager.preferredLayoutSize(Container)](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container)) is used to calculate the preferred size.

Note: some implementations may cache the value returned from the LayoutManager. Implementations that cache need not invoke preferredLayoutSize on the LayoutManager every time this method is invoked, rather the LayoutManager will only be queried after the Container becomes invalid.

**Overrides:**[getPreferredSize](http://docs.google.com/java/awt/Component.html#getPreferredSize()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**an instance of Dimension that represents the preferred size of this container.**See Also:**[getMinimumSize()](http://docs.google.com/java/awt/Container.html#getMinimumSize()), [getMaximumSize()](http://docs.google.com/java/awt/Container.html#getMaximumSize()), [getLayout()](http://docs.google.com/java/awt/Container.html#getLayout()), [LayoutManager.preferredLayoutSize(Container)](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container)), [Component.getPreferredSize()](http://docs.google.com/java/awt/Component.html#getPreferredSize())

### preferredSize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Dimension](http://docs.google.com/java/awt/Dimension.html) **preferredSize**()

**Deprecated.** *As of JDK version 1.1, replaced by getPreferredSize().*

**Overrides:**[preferredSize](http://docs.google.com/java/awt/Component.html#preferredSize()) in class [Component](http://docs.google.com/java/awt/Component.html)

### getMinimumSize

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

Returns the minimum size of this container. If the minimum size has not been set explicitly by [Component.setMinimumSize(Dimension)](http://docs.google.com/java/awt/Component.html#setMinimumSize(java.awt.Dimension)) and this Container has a non-null [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), then [LayoutManager.minimumLayoutSize(Container)](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)) is used to calculate the minimum size.

Note: some implementations may cache the value returned from the LayoutManager. Implementations that cache need not invoke minimumLayoutSize on the LayoutManager every time this method is invoked, rather the LayoutManager will only be queried after the Container becomes invalid.

**Overrides:**[getMinimumSize](http://docs.google.com/java/awt/Component.html#getMinimumSize()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**an instance of Dimension that represents the minimum size of this container.**Since:** JDK1.1 **See Also:**[getPreferredSize()](http://docs.google.com/java/awt/Container.html#getPreferredSize()), [getMaximumSize()](http://docs.google.com/java/awt/Container.html#getMaximumSize()), [getLayout()](http://docs.google.com/java/awt/Container.html#getLayout()), [LayoutManager.minimumLayoutSize(Container)](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)), [Component.getMinimumSize()](http://docs.google.com/java/awt/Component.html#getMinimumSize())

### minimumSize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Dimension](http://docs.google.com/java/awt/Dimension.html) **minimumSize**()

**Deprecated.** *As of JDK version 1.1, replaced by getMinimumSize().*

**Overrides:**[minimumSize](http://docs.google.com/java/awt/Component.html#minimumSize()) in class [Component](http://docs.google.com/java/awt/Component.html)

### getMaximumSize

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

Returns the maximum size of this container. If the maximum size has not been set explicitly by [Component.setMaximumSize(Dimension)](http://docs.google.com/java/awt/Component.html#setMaximumSize(java.awt.Dimension)) and the [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) installed on this Container is an instance of [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html), then [LayoutManager2.maximumLayoutSize(Container)](http://docs.google.com/java/awt/LayoutManager2.html#maximumLayoutSize(java.awt.Container)) is used to calculate the maximum size.

Note: some implementations may cache the value returned from the LayoutManager2. Implementations that cache need not invoke maximumLayoutSize on the LayoutManager2 every time this method is invoked, rather the LayoutManager2 will only be queried after the Container becomes invalid.

**Overrides:**[getMaximumSize](http://docs.google.com/java/awt/Component.html#getMaximumSize()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**an instance of Dimension that represents the maximum size of this container.**See Also:**[getPreferredSize()](http://docs.google.com/java/awt/Container.html#getPreferredSize()), [getMinimumSize()](http://docs.google.com/java/awt/Container.html#getMinimumSize()), [getLayout()](http://docs.google.com/java/awt/Container.html#getLayout()), [LayoutManager2.maximumLayoutSize(Container)](http://docs.google.com/java/awt/LayoutManager2.html#maximumLayoutSize(java.awt.Container)), [Component.getMaximumSize()](http://docs.google.com/java/awt/Component.html#getMaximumSize())

### getAlignmentX

public float **getAlignmentX**()

Returns the alignment along the x axis. This specifies how the component would like to be aligned relative to other components. The value should be a number between 0 and 1 where 0 represents alignment along the origin, 1 is aligned the furthest away from the origin, 0.5 is centered, etc.

**Overrides:**[getAlignmentX](http://docs.google.com/java/awt/Component.html#getAlignmentX()) in class [Component](http://docs.google.com/java/awt/Component.html)

### getAlignmentY

public float **getAlignmentY**()

Returns the alignment along the y axis. This specifies how the component would like to be aligned relative to other components. The value should be a number between 0 and 1 where 0 represents alignment along the origin, 1 is aligned the furthest away from the origin, 0.5 is centered, etc.

**Overrides:**[getAlignmentY](http://docs.google.com/java/awt/Component.html#getAlignmentY()) in class [Component](http://docs.google.com/java/awt/Component.html)

### paint

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

Paints the container. This forwards the paint to any lightweight components that are children of this container. If this method is reimplemented, super.paint(g) should be called so that lightweight components are properly rendered. If a child component is entirely clipped by the current clipping setting in g, paint() will not be forwarded to that child.

**Overrides:**[paint](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**g - the specified Graphics window**See Also:**[Component.update(Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### update

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

Updates the container. This forwards the update to any lightweight components that are children of this container. If this method is reimplemented, super.update(g) should be called so that lightweight components are properly rendered. If a child component is entirely clipped by the current clipping setting in g, update() will not be forwarded to that child.

**Overrides:**[update](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**g - the specified Graphics window**See Also:**[Component.update(Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### print

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

Prints the container. This forwards the print to any lightweight components that are children of this container. If this method is reimplemented, super.print(g) should be called so that lightweight components are properly rendered. If a child component is entirely clipped by the current clipping setting in g, print() will not be forwarded to that child.

**Overrides:**[print](http://docs.google.com/java/awt/Component.html#print(java.awt.Graphics)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**g - the specified Graphics window**See Also:**[Component.update(Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### paintComponents

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

Paints each of the components in this container.

**Parameters:**g - the graphics context.**See Also:**[Component.paint(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics)), [Component.paintAll(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics))

### printComponents

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

Prints each of the components in this container.

**Parameters:**g - the graphics context.**See Also:**[Component.print(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#print(java.awt.Graphics)), [Component.printAll(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#printAll(java.awt.Graphics))

### addContainerListener

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

Adds the specified container listener to receive container events from this container. 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.

**Parameters:**l - the container listener**See Also:**[removeContainerListener(java.awt.event.ContainerListener)](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener)), [getContainerListeners()](http://docs.google.com/java/awt/Container.html#getContainerListeners())

### removeContainerListener

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

Removes the specified container listener so it no longer receives container events from this container. 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.

**Parameters:**l - the container listener**See Also:**[addContainerListener(java.awt.event.ContainerListener)](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener)), [getContainerListeners()](http://docs.google.com/java/awt/Container.html#getContainerListeners())

### getContainerListeners

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

Returns an array of all the container listeners registered on this container.

**Returns:**all of this container's ContainerListeners or an empty array if no container listeners are currently registered**Since:** 1.4 **See Also:**[addContainerListener(java.awt.event.ContainerListener)](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener)), [removeContainerListener(java.awt.event.ContainerListener)](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener))

### 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 Container. *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 Container c for its container listeners with the following code:

ContainerListener[] cls = (ContainerListener[])(c.getListeners(ContainerListener.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 container, 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:**[getContainerListeners()](http://docs.google.com/java/awt/Container.html#getContainerListeners())

### processEvent

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

Processes events on this container. If the event is a ContainerEvent, it invokes the processContainerEvent method, else it invokes its superclass's processEvent.

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**See Also:**[Component.processComponentEvent(java.awt.event.ComponentEvent)](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [Component.processFocusEvent(java.awt.event.FocusEvent)](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [Component.processKeyEvent(java.awt.event.KeyEvent)](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent)), [Component.processMouseEvent(java.awt.event.MouseEvent)](http://docs.google.com/java/awt/Component.html#processMouseEvent(java.awt.event.MouseEvent)), [Component.processMouseMotionEvent(java.awt.event.MouseEvent)](http://docs.google.com/java/awt/Component.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [Component.processInputMethodEvent(java.awt.event.InputMethodEvent)](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [Component.processHierarchyEvent(java.awt.event.HierarchyEvent)](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [Component.processMouseWheelEvent(java.awt.event.MouseWheelEvent)](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent))

### processContainerEvent

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

Processes container events occurring on this container by dispatching them to any registered ContainerListener objects. NOTE: This method will not be called unless container events are enabled for this component; this happens when one of the following occurs:

* A ContainerListener object is registered via addContainerListener
* Container 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 container event**See Also:**[Component.enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### deliverEvent

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **deliverEvent**([Event](http://docs.google.com/java/awt/Event.html) e)

**Deprecated.** *As of JDK version 1.1, replaced by dispatchEvent(AWTEvent e)*

**Overrides:**[deliverEvent](http://docs.google.com/java/awt/Component.html#deliverEvent(java.awt.Event)) in class [Component](http://docs.google.com/java/awt/Component.html)

### getComponentAt

public [Component](http://docs.google.com/java/awt/Component.html) **getComponentAt**(int x,  
 int y)

Locates the component that contains the x,y position. The top-most child component is returned in the case where there is overlap in the components. This is determined by finding the component closest to the index 0 that claims to contain the given point via Component.contains(), except that Components which have native peers take precedence over those which do not (i.e., lightweight Components).

**Overrides:**[getComponentAt](http://docs.google.com/java/awt/Component.html#getComponentAt(int,%20int)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**x - the *x* coordinatey - the *y* coordinate **Returns:**null if the component does not contain the position. If there is no child component at the requested point and the point is within the bounds of the container the container itself is returned; otherwise the top-most child is returned.**Since:** JDK1.1 **See Also:**[Component.contains(int, int)](http://docs.google.com/java/awt/Component.html#contains(int,%20int))

### locate

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Component](http://docs.google.com/java/awt/Component.html) **locate**(int x,  
 int y)

**Deprecated.** *As of JDK version 1.1, replaced by getComponentAt(int, int).*

**Overrides:**[locate](http://docs.google.com/java/awt/Component.html#locate(int,%20int)) in class [Component](http://docs.google.com/java/awt/Component.html)

### getComponentAt

public [Component](http://docs.google.com/java/awt/Component.html) **getComponentAt**([Point](http://docs.google.com/java/awt/Point.html) p)

Gets the component that contains the specified point.

**Overrides:**[getComponentAt](http://docs.google.com/java/awt/Component.html#getComponentAt(java.awt.Point)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**p - the point. **Returns:**returns the component that contains the point, or null if the component does not contain the point.**Since:** JDK1.1 **See Also:**[Component.contains(int, int)](http://docs.google.com/java/awt/Component.html#contains(int,%20int))

### getMousePosition

public [Point](http://docs.google.com/java/awt/Point.html) **getMousePosition**(boolean allowChildren)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Returns the position of the mouse pointer in this Container's coordinate space if the Container is under the mouse pointer, otherwise returns null. This method is similar to [Component.getMousePosition()](http://docs.google.com/java/awt/Component.html#getMousePosition()) with the exception that it can take the Container's children into account. If allowChildren is false, this method will return a non-null value only if the mouse pointer is above the Container directly, not above the part obscured by children. If allowChildren is true, this method returns a non-null value if the mouse pointer is above Container or any of its descendants.

**Parameters:**allowChildren - true if children should be taken into account **Returns:**mouse coordinates relative to this Component, or null **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**Since:** 1.5 **See Also:**[Component.getMousePosition()](http://docs.google.com/java/awt/Component.html#getMousePosition())

### findComponentAt

public [Component](http://docs.google.com/java/awt/Component.html) **findComponentAt**(int x,  
 int y)

Locates the visible child component that contains the specified position. The top-most child component is returned in the case where there is overlap in the components. If the containing child component is a Container, this method will continue searching for the deepest nested child component. Components which are not visible are ignored during the search.

The findComponentAt method is different from getComponentAt in that getComponentAt only searches the Container's immediate children; if the containing component is a Container, findComponentAt will search that child to find a nested component.

**Parameters:**x - the *x* coordinatey - the *y* coordinate **Returns:**null if the component does not contain the position. If there is no child component at the requested point and the point is within the bounds of the container the container itself is returned.**Since:** 1.2 **See Also:**[Component.contains(int, int)](http://docs.google.com/java/awt/Component.html#contains(int,%20int)), [getComponentAt(int, int)](http://docs.google.com/java/awt/Container.html#getComponentAt(int,%20int))

### findComponentAt

public [Component](http://docs.google.com/java/awt/Component.html) **findComponentAt**([Point](http://docs.google.com/java/awt/Point.html) p)

Locates the visible child component that contains the specified point. The top-most child component is returned in the case where there is overlap in the components. If the containing child component is a Container, this method will continue searching for the deepest nested child component. Components which are not visible are ignored during the search.

The findComponentAt method is different from getComponentAt in that getComponentAt only searches the Container's immediate children; if the containing component is a Container, findComponentAt will search that child to find a nested component.

**Parameters:**p - the point. **Returns:**null if the component does not contain the position. If there is no child component at the requested point and the point is within the bounds of the container the container itself is returned.**Since:** 1.2 **See Also:**[Component.contains(int, int)](http://docs.google.com/java/awt/Component.html#contains(int,%20int)), [getComponentAt(int, int)](http://docs.google.com/java/awt/Container.html#getComponentAt(int,%20int))

### addNotify

public void **addNotify**()

Makes this Container displayable by connecting it to a native screen resource. Making a container displayable will cause all of its children to be made displayable. This method is called internally by the toolkit and should not be called directly by programs.

**Overrides:**[addNotify](http://docs.google.com/java/awt/Component.html#addNotify()) in class [Component](http://docs.google.com/java/awt/Component.html) **See Also:**[Component.isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [removeNotify()](http://docs.google.com/java/awt/Container.html#removeNotify())

### removeNotify

public void **removeNotify**()

Makes this Container undisplayable by removing its connection to its native screen resource. Making a container undisplayable will cause all of its children to be made undisplayable. This method is called by the toolkit internally and should not be called directly by programs.

**Overrides:**[removeNotify](http://docs.google.com/java/awt/Component.html#removeNotify()) in class [Component](http://docs.google.com/java/awt/Component.html) **See Also:**[Component.isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [addNotify()](http://docs.google.com/java/awt/Container.html#addNotify())

### isAncestorOf

public boolean **isAncestorOf**([Component](http://docs.google.com/java/awt/Component.html) c)

Checks if the component is contained in the component hierarchy of this container.

**Parameters:**c - the component **Returns:**true if it is an ancestor; false otherwise.**Since:** JDK1.1

### paramString

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

Returns a string representing the state of this Container. 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 container

### list

public void **list**([PrintStream](http://docs.google.com/java/io/PrintStream.html) out,  
 int indent)

Prints a listing of this container to the specified output stream. The listing starts at the specified indentation.

The immediate children of the container are printed with an indentation of indent+1. The children of those children are printed at indent+2 and so on.

**Overrides:**[list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream,%20int)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**out - a print streamindent - the number of spaces to indent**Since:** JDK1.0 **See Also:**[Component.list(java.io.PrintStream, int)](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream,%20int))

### list

public void **list**([PrintWriter](http://docs.google.com/java/io/PrintWriter.html) out,  
 int indent)

Prints out a list, starting at the specified indentation, to the specified print writer.

The immediate children of the container are printed with an indentation of indent+1. The children of those children are printed at indent+2 and so on.

**Overrides:**[list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter,%20int)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**out - a print writerindent - the number of spaces to indent**Since:** JDK1.1 **See Also:**[Component.list(java.io.PrintWriter, int)](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter,%20int))

### setFocusTraversalKeys

public void **setFocusTraversalKeys**(int id,  
 [Set](http://docs.google.com/java/util/Set.html)<? extends [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html)> keystrokes)

Sets the focus traversal keys for a given traversal operation for this Container.

The default values for a Container's focus traversal keys are implementation-dependent. Sun recommends that all implementations for a particular native platform use the same default values. The recommendations for Windows and Unix are listed below. These recommendations are used in the Sun AWT implementations.

| Identifier | Meaning | Default |
| --- | --- | --- |
| KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS | Normal forward keyboard traversal | TAB on KEY\_PRESSED, CTRL-TAB on KEY\_PRESSED |
| KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS | Normal reverse keyboard traversal | SHIFT-TAB on KEY\_PRESSED, CTRL-SHIFT-TAB on KEY\_PRESSED |
| KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS | Go up one focus traversal cycle | none |
| KeyboardFocusManager.DOWN\_CYCLE\_TRAVERSAL\_KEYS |  | Go down one focus traversal cycle | none |

To disable a traversal key, use an empty Set; Collections.EMPTY\_SET is recommended.

Using the AWTKeyStroke API, client code can specify on which of two specific KeyEvents, KEY\_PRESSED or KEY\_RELEASED, the focus traversal operation will occur. Regardless of which KeyEvent is specified, however, all KeyEvents related to the focus traversal key, including the associated KEY\_TYPED event, will be consumed, and will not be dispatched to any Container. It is a runtime error to specify a KEY\_TYPED event as mapping to a focus traversal operation, or to map the same event to multiple default focus traversal operations.

If a value of null is specified for the Set, this Container inherits the Set from its parent. If all ancestors of this Container have null specified for the Set, then the current KeyboardFocusManager's default Set is used.

**Overrides:**[setFocusTraversalKeys](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeys(int,%20java.util.Set)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**id - one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS, or KeyboardFocusManager.DOWN\_CYCLE\_TRAVERSAL\_KEYSkeystrokes - the Set of AWTKeyStroke for the specified operation **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if id is not one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS, or KeyboardFocusManager.DOWN\_CYCLE\_TRAVERSAL\_KEYS, or if keystrokes contains null, or if any Object in keystrokes is not an AWTKeyStroke, or if any keystroke represents a KEY\_TYPED event, or if any keystroke already maps to another focus traversal operation for this Container**Since:** 1.4 **See Also:**[getFocusTraversalKeys(int)](http://docs.google.com/java/awt/Container.html#getFocusTraversalKeys(int)), [KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#FORWARD_TRAVERSAL_KEYS), [KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#BACKWARD_TRAVERSAL_KEYS), [KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#UP_CYCLE_TRAVERSAL_KEYS), [KeyboardFocusManager.DOWN\_CYCLE\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#DOWN_CYCLE_TRAVERSAL_KEYS)

### getFocusTraversalKeys

public [Set](http://docs.google.com/java/util/Set.html)<[AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html)> **getFocusTraversalKeys**(int id)

Returns the Set of focus traversal keys for a given traversal operation for this Container. (See setFocusTraversalKeys for a full description of each key.)

If a Set of traversal keys has not been explicitly defined for this Container, then this Container's parent's Set is returned. If no Set has been explicitly defined for any of this Container's ancestors, then the current KeyboardFocusManager's default Set is returned.

**Overrides:**[getFocusTraversalKeys](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeys(int)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**id - one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS, or KeyboardFocusManager.DOWN\_CYCLE\_TRAVERSAL\_KEYS **Returns:**the Set of AWTKeyStrokes for the specified operation. The Set will be unmodifiable, and may be empty. null will never be returned. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if id is not one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS, or KeyboardFocusManager.DOWN\_CYCLE\_TRAVERSAL\_KEYS**Since:** 1.4 **See Also:**[setFocusTraversalKeys(int, java.util.Set)](http://docs.google.com/java/awt/Container.html#setFocusTraversalKeys(int,%20java.util.Set)), [KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#FORWARD_TRAVERSAL_KEYS), [KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#BACKWARD_TRAVERSAL_KEYS), [KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#UP_CYCLE_TRAVERSAL_KEYS), [KeyboardFocusManager.DOWN\_CYCLE\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#DOWN_CYCLE_TRAVERSAL_KEYS)

### areFocusTraversalKeysSet

public boolean **areFocusTraversalKeysSet**(int id)

Returns whether the Set of focus traversal keys for the given focus traversal operation has been explicitly defined for this Container. If this method returns false, this Container is inheriting the Set from an ancestor, or from the current KeyboardFocusManager.

**Overrides:**[areFocusTraversalKeysSet](http://docs.google.com/java/awt/Component.html#areFocusTraversalKeysSet(int)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**id - one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS, or KeyboardFocusManager.DOWN\_CYCLE\_TRAVERSAL\_KEYS **Returns:**true if the the Set of focus traversal keys for the given focus traversal operation has been explicitly defined for this Component; false otherwise. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if id is not one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS, or KeyboardFocusManager.DOWN\_CYCLE\_TRAVERSAL\_KEYS**Since:** 1.4

### isFocusCycleRoot

public boolean **isFocusCycleRoot**([Container](http://docs.google.com/java/awt/Container.html) container)

Returns whether the specified Container is the focus cycle root of this Container's focus traversal cycle. Each focus traversal cycle has only a single focus cycle root and each Container which is not a focus cycle root belongs to only a single focus traversal cycle. Containers which are focus cycle roots belong to two cycles: one rooted at the Container itself, and one rooted at the Container's nearest focus-cycle-root ancestor. This method will return true for both such Containers in this case.

**Overrides:**[isFocusCycleRoot](http://docs.google.com/java/awt/Component.html#isFocusCycleRoot(java.awt.Container)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**container - the Container to be tested **Returns:**true if the specified Container is a focus-cycle- root of this Container; false otherwise**Since:** 1.4 **See Also:**[isFocusCycleRoot()](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot())

### transferFocusBackward

public void **transferFocusBackward**()

**Description copied from class:** [**Component**](http://docs.google.com/java/awt/Component.html#transferFocusBackward()) Transfers the focus to the previous component, as though this Component were the focus owner.

**Overrides:**[transferFocusBackward](http://docs.google.com/java/awt/Component.html#transferFocusBackward()) in class [Component](http://docs.google.com/java/awt/Component.html) **See Also:**[Component.requestFocus()](http://docs.google.com/java/awt/Component.html#requestFocus())

### setFocusTraversalPolicy

public void **setFocusTraversalPolicy**([FocusTraversalPolicy](http://docs.google.com/java/awt/FocusTraversalPolicy.html) policy)

Sets the focus traversal policy that will manage keyboard traversal of this Container's children, if this Container is a focus cycle root. If the argument is null, this Container inherits its policy from its focus- cycle-root ancestor. If the argument is non-null, this policy will be inherited by all focus-cycle-root children that have no keyboard- traversal policy of their own (as will, recursively, their focus-cycle- root children).

If this Container is not a focus cycle root, the policy will be remembered, but will not be used or inherited by this or any other Containers until this Container is made a focus cycle root.

**Parameters:**policy - the new focus traversal policy for this Container**Since:** 1.4 **See Also:**[getFocusTraversalPolicy()](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [setFocusCycleRoot(boolean)](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [isFocusCycleRoot(java.awt.Container)](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container))

### getFocusTraversalPolicy

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

Returns the focus traversal policy that will manage keyboard traversal of this Container's children, or null if this Container is not a focus cycle root. If no traversal policy has been explicitly set for this Container, then this Container's focus-cycle-root ancestor's policy is returned.

**Returns:**this Container's focus traversal policy, or null if this Container is not a focus cycle root.**Since:** 1.4 **See Also:**[setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusCycleRoot(boolean)](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [isFocusCycleRoot(java.awt.Container)](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container))

### isFocusTraversalPolicySet

public boolean **isFocusTraversalPolicySet**()

Returns whether the focus traversal policy has been explicitly set for this Container. If this method returns false, this Container will inherit its focus traversal policy from an ancestor.

**Returns:**true if the focus traversal policy has been explicitly set for this Container; false otherwise.**Since:** 1.4

### setFocusCycleRoot

public void **setFocusCycleRoot**(boolean focusCycleRoot)

Sets whether this Container is the root of a focus traversal cycle. Once focus enters a traversal cycle, typically it cannot leave it via focus traversal unless one of the up- or down-cycle keys is pressed. Normal traversal is limited to this Container, and all of this Container's descendants that are not descendants of inferior focus cycle roots. Note that a FocusTraversalPolicy may bend these restrictions, however. For example, ContainerOrderFocusTraversalPolicy supports implicit down-cycle traversal.

The alternative way to specify the traversal order of this Container's children is to make this Container a [focus traversal policy provider](http://docs.google.com/doc-files/FocusSpec.html#FocusTraversalPolicyProviders).

**Parameters:**focusCycleRoot - indicates whether this Container is the root of a focus traversal cycle**Since:** 1.4 **See Also:**[isFocusCycleRoot()](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot()), [setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [getFocusTraversalPolicy()](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [ContainerOrderFocusTraversalPolicy](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html), [setFocusTraversalPolicyProvider(boolean)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean))

### isFocusCycleRoot

public boolean **isFocusCycleRoot**()

Returns whether this Container is the root of a focus traversal cycle. Once focus enters a traversal cycle, typically it cannot leave it via focus traversal unless one of the up- or down-cycle keys is pressed. Normal traversal is limited to this Container, and all of this Container's descendants that are not descendants of inferior focus cycle roots. Note that a FocusTraversalPolicy may bend these restrictions, however. For example, ContainerOrderFocusTraversalPolicy supports implicit down-cycle traversal.

**Returns:**whether this Container is the root of a focus traversal cycle**Since:** 1.4 **See Also:**[setFocusCycleRoot(boolean)](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [getFocusTraversalPolicy()](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [ContainerOrderFocusTraversalPolicy](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html)

### setFocusTraversalPolicyProvider

public final void **setFocusTraversalPolicyProvider**(boolean provider)

Sets whether this container will be used to provide focus traversal policy. Container with this property as true will be used to acquire focus traversal policy instead of closest focus cycle root ancestor.

**Parameters:**provider - indicates whether this container will be used to provide focus traversal policy**Since:** 1.5 **See Also:**[setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [getFocusTraversalPolicy()](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [isFocusTraversalPolicyProvider()](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicyProvider())

### isFocusTraversalPolicyProvider

public final boolean **isFocusTraversalPolicyProvider**()

Returns whether this container provides focus traversal policy. If this property is set to true then when keyboard focus manager searches container hierarchy for focus traversal policy and encounters this container before any other container with this property as true or focus cycle roots then its focus traversal policy will be used instead of focus cycle root's policy.

**Returns:**true if this container provides focus traversal policy, false otherwise**Since:** 1.5 **See Also:**[setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [getFocusTraversalPolicy()](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [setFocusCycleRoot(boolean)](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [setFocusTraversalPolicyProvider(boolean)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean))

### transferFocusDownCycle

public void **transferFocusDownCycle**()

Transfers the focus down one focus traversal cycle. If this Container is a focus cycle root, then the focus owner is set to this Container's default Component to focus, and the current focus cycle root is set to this Container. If this Container is not a focus cycle root, then no focus traversal operation occurs.

**Since:** 1.4 **See Also:**[Component.requestFocus()](http://docs.google.com/java/awt/Component.html#requestFocus()), [isFocusCycleRoot(java.awt.Container)](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container)), [setFocusCycleRoot(boolean)](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean))

### applyComponentOrientation

public void **applyComponentOrientation**([ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html) o)

Sets the ComponentOrientation property of this container and all components contained within it.

**Overrides:**[applyComponentOrientation](http://docs.google.com/java/awt/Component.html#applyComponentOrientation(java.awt.ComponentOrientation)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**o - the new component orientation of this container and the components contained within it. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if orientation is null.**Since:** 1.4 **See Also:**[Component.setComponentOrientation(java.awt.ComponentOrientation)](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [Component.getComponentOrientation()](http://docs.google.com/java/awt/Component.html#getComponentOrientation())

### addPropertyChangeListener

public void **addPropertyChangeListener**([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Adds a PropertyChangeListener to the listener list. The listener is registered for all bound properties of this class, including the following:

* this Container's font ("font")
* this Container's background color ("background")
* this Container's foreground color ("foreground")
* this Container's focusability ("focusable")
* this Container's focus traversal keys enabled state ("focusTraversalKeysEnabled")
* this Container's Set of FORWARD\_TRAVERSAL\_KEYS ("forwardFocusTraversalKeys")
* this Container's Set of BACKWARD\_TRAVERSAL\_KEYS ("backwardFocusTraversalKeys")
* this Container's Set of UP\_CYCLE\_TRAVERSAL\_KEYS ("upCycleFocusTraversalKeys")
* this Container's Set of DOWN\_CYCLE\_TRAVERSAL\_KEYS ("downCycleFocusTraversalKeys")
* this Container's focus traversal policy ("focusTraversalPolicy")
* this Container's focus-cycle-root state ("focusCycleRoot")

Note that if this Container is inheriting a bound property, then no event will be fired in response to a change in the inherited property.

If listener is null, no exception is thrown and no action is performed.

**Overrides:**[addPropertyChangeListener](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.beans.PropertyChangeListener)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**listener - the PropertyChangeListener to be added**See Also:**[Component.removePropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener(java.lang.String,java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener))

### addPropertyChangeListener

public void **addPropertyChangeListener**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Adds a PropertyChangeListener to the listener list for a specific property. The specified property may be user-defined, or one of the following defaults:

* this Container's font ("font")
* this Container's background color ("background")
* this Container's foreground color ("foreground")
* this Container's focusability ("focusable")
* this Container's focus traversal keys enabled state ("focusTraversalKeysEnabled")
* this Container's Set of FORWARD\_TRAVERSAL\_KEYS ("forwardFocusTraversalKeys")
* this Container's Set of BACKWARD\_TRAVERSAL\_KEYS ("backwardFocusTraversalKeys")
* this Container's Set of UP\_CYCLE\_TRAVERSAL\_KEYS ("upCycleFocusTraversalKeys")
* this Container's Set of DOWN\_CYCLE\_TRAVERSAL\_KEYS ("downCycleFocusTraversalKeys")
* this Container's focus traversal policy ("focusTraversalPolicy")
* this Container's focus-cycle-root state ("focusCycleRoot")
* this Container's focus-traversal-policy-provider state("focusTraversalPolicyProvider")
* this Container's focus-traversal-policy-provider state("focusTraversalPolicyProvider")

Note that if this Container is inheriting a bound property, then no event will be fired in response to a change in the inherited property.

If listener is null, no exception is thrown and no action is performed.

**Overrides:**[addPropertyChangeListener](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**propertyName - one of the property names listed abovelistener - the PropertyChangeListener to be added**See Also:**[addPropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [Component.removePropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Container.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/CompositeContext.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Container.AccessibleAWTContainer.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Container.html)    [**NO FRAMES**](http://docs.google.com/Container.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: FIELD | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

[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).