| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JComboBox.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/javax/swing/JColorChooser.AccessibleJColorChooser.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JComboBox.AccessibleJComboBox.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JComboBox.html)    [**NO FRAMES**](http://docs.google.com/JComboBox.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#3as4poj) | [METHOD](#3o7alnk) |

## **javax.swing**

Class JComboBox

[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](http://docs.google.com/java/awt/Container.html)  
 [javax.swing.JComponent](http://docs.google.com/javax/swing/JComponent.html)  
 **javax.swing.JComboBox**

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

public class **JComboBox**extends [JComponent](http://docs.google.com/javax/swing/JComponent.html)implements [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html), [ListDataListener](http://docs.google.com/javax/swing/event/ListDataListener.html), [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

A component that combines a button or editable field and a drop-down list. The user can select a value from the drop-down list, which appears at the user's request. If you make the combo box editable, then the combo box includes an editable field into which the user can type a value.

**Warning:** Swing is not thread safe. For more information see [Swing's Threading Policy](http://docs.google.com/package-summary.html#threading).

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

See [How to Use Combo Boxes](http://java.sun.com/docs/books/tutorial/uiswing/components/combobox.html) in [*The Java Tutorial*](http://java.sun.com/Series/Tutorial/index.html) for further information.

**See Also:**[ComboBoxModel](http://docs.google.com/javax/swing/ComboBoxModel.html), [DefaultComboBoxModel](http://docs.google.com/javax/swing/DefaultComboBoxModel.html)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JComboBox.AccessibleJComboBox**](http://docs.google.com/javax/swing/JComboBox.AccessibleJComboBox.html)            This class implements accessibility support for the JComboBox class. |
| static interface | [**JComboBox.KeySelectionManager**](http://docs.google.com/javax/swing/JComboBox.KeySelectionManager.html)            The interface that defines a KeySelectionManager. |

| **Nested classes/interfaces inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [JComponent.AccessibleJComponent](http://docs.google.com/javax/swing/JComponent.AccessibleJComponent.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [Container.AccessibleAWTContainer](http://docs.google.com/java/awt/Container.AccessibleAWTContainer.html) |

| **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** | |
| --- | --- |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**actionCommand**](http://docs.google.com/javax/swing/JComboBox.html#actionCommand)            This protected field is implementation specific. |
| protected  [ComboBoxModel](http://docs.google.com/javax/swing/ComboBoxModel.html) | [**dataModel**](http://docs.google.com/javax/swing/JComboBox.html#dataModel)            This protected field is implementation specific. |
| protected  [ComboBoxEditor](http://docs.google.com/javax/swing/ComboBoxEditor.html) | [**editor**](http://docs.google.com/javax/swing/JComboBox.html#editor)            This protected field is implementation specific. |
| protected  boolean | [**isEditable**](http://docs.google.com/javax/swing/JComboBox.html#isEditable)            This protected field is implementation specific. |
| protected  [JComboBox.KeySelectionManager](http://docs.google.com/javax/swing/JComboBox.KeySelectionManager.html) | [**keySelectionManager**](http://docs.google.com/javax/swing/JComboBox.html#keySelectionManager)            This protected field is implementation specific. |
| protected  boolean | [**lightWeightPopupEnabled**](http://docs.google.com/javax/swing/JComboBox.html#lightWeightPopupEnabled)            This protected field is implementation specific. |
| protected  int | [**maximumRowCount**](http://docs.google.com/javax/swing/JComboBox.html#maximumRowCount)            This protected field is implementation specific. |
| protected  [ListCellRenderer](http://docs.google.com/javax/swing/ListCellRenderer.html) | [**renderer**](http://docs.google.com/javax/swing/JComboBox.html#renderer)            This protected field is implementation specific. |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**selectedItemReminder**](http://docs.google.com/javax/swing/JComboBox.html#selectedItemReminder)            This protected field is implementation specific. |

| **Fields inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [accessibleContext](http://docs.google.com/javax/swing/JComponent.html#accessibleContext), [listenerList](http://docs.google.com/javax/swing/JComponent.html#listenerList), [TOOL\_TIP\_TEXT\_KEY](http://docs.google.com/javax/swing/JComponent.html#TOOL_TIP_TEXT_KEY), [ui](http://docs.google.com/javax/swing/JComponent.html#ui), [UNDEFINED\_CONDITION](http://docs.google.com/javax/swing/JComponent.html#UNDEFINED_CONDITION), [WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT](http://docs.google.com/javax/swing/JComponent.html#WHEN_ANCESTOR_OF_FOCUSED_COMPONENT), [WHEN\_FOCUSED](http://docs.google.com/javax/swing/JComponent.html#WHEN_FOCUSED), [WHEN\_IN\_FOCUSED\_WINDOW](http://docs.google.com/javax/swing/JComponent.html#WHEN_IN_FOCUSED_WINDOW) |

| **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** | |
| --- | --- |
| [**JComboBox**](http://docs.google.com/javax/swing/JComboBox.html#JComboBox())()            Creates a JComboBox with a default data model. |
| [**JComboBox**](http://docs.google.com/javax/swing/JComboBox.html#JComboBox(javax.swing.ComboBoxModel))([ComboBoxModel](http://docs.google.com/javax/swing/ComboBoxModel.html) aModel)            Creates a JComboBox that takes its items from an existing ComboBoxModel. |
| [**JComboBox**](http://docs.google.com/javax/swing/JComboBox.html#JComboBox(java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html)[] items)            Creates a JComboBox that contains the elements in the specified array. |
| [**JComboBox**](http://docs.google.com/javax/swing/JComboBox.html#JComboBox(java.util.Vector))([Vector](http://docs.google.com/java/util/Vector.html)<?> items)            Creates a JComboBox that contains the elements in the specified Vector. |

| **Method Summary** | |
| --- | --- |
| void | [**actionPerformed**](http://docs.google.com/javax/swing/JComboBox.html#actionPerformed(java.awt.event.ActionEvent))([ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html) e)            This method is public as an implementation side effect. |
| protected  void | [**actionPropertyChanged**](http://docs.google.com/javax/swing/JComboBox.html#actionPropertyChanged(javax.swing.Action,%20java.lang.String))([Action](http://docs.google.com/javax/swing/Action.html) action, [String](http://docs.google.com/java/lang/String.html) propertyName)            Updates the combobox's state in response to property changes in associated action. |
| void | [**addActionListener**](http://docs.google.com/javax/swing/JComboBox.html#addActionListener(java.awt.event.ActionListener))([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)            Adds an ActionListener. |
| void | [**addItem**](http://docs.google.com/javax/swing/JComboBox.html#addItem(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) anObject)            Adds an item to the item list. |
| void | [**addItemListener**](http://docs.google.com/javax/swing/JComboBox.html#addItemListener(java.awt.event.ItemListener))([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) aListener)            Adds an ItemListener. |
| void | [**addPopupMenuListener**](http://docs.google.com/javax/swing/JComboBox.html#addPopupMenuListener(javax.swing.event.PopupMenuListener))([PopupMenuListener](http://docs.google.com/javax/swing/event/PopupMenuListener.html) l)            Adds a PopupMenu listener which will listen to notification messages from the popup portion of the combo box. |
| void | [**configureEditor**](http://docs.google.com/javax/swing/JComboBox.html#configureEditor(javax.swing.ComboBoxEditor,%20java.lang.Object))([ComboBoxEditor](http://docs.google.com/javax/swing/ComboBoxEditor.html) anEditor, [Object](http://docs.google.com/java/lang/Object.html) anItem)            Initializes the editor with the specified item. |
| protected  void | [**configurePropertiesFromAction**](http://docs.google.com/javax/swing/JComboBox.html#configurePropertiesFromAction(javax.swing.Action))([Action](http://docs.google.com/javax/swing/Action.html) a)            Sets the properties on this combobox to match those in the specified Action. |
| void | [**contentsChanged**](http://docs.google.com/javax/swing/JComboBox.html#contentsChanged(javax.swing.event.ListDataEvent))([ListDataEvent](http://docs.google.com/javax/swing/event/ListDataEvent.html) e)            This method is public as an implementation side effect. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createActionPropertyChangeListener**](http://docs.google.com/javax/swing/JComboBox.html#createActionPropertyChangeListener(javax.swing.Action))([Action](http://docs.google.com/javax/swing/Action.html) a)            Creates and returns a PropertyChangeListener that is responsible for listening for changes from the specified Action and updating the appropriate properties. |
| protected  [JComboBox.KeySelectionManager](http://docs.google.com/javax/swing/JComboBox.KeySelectionManager.html) | [**createDefaultKeySelectionManager**](http://docs.google.com/javax/swing/JComboBox.html#createDefaultKeySelectionManager())()            Returns an instance of the default key-selection manager. |
| protected  void | [**fireActionEvent**](http://docs.google.com/javax/swing/JComboBox.html#fireActionEvent())()            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireItemStateChanged**](http://docs.google.com/javax/swing/JComboBox.html#fireItemStateChanged(java.awt.event.ItemEvent))([ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html) e)            Notifies all listeners that have registered interest for notification on this event type. |
| void | [**firePopupMenuCanceled**](http://docs.google.com/javax/swing/JComboBox.html#firePopupMenuCanceled())()            Notifies PopupMenuListeners that the popup portion of the combo box has been canceled. |
| void | [**firePopupMenuWillBecomeInvisible**](http://docs.google.com/javax/swing/JComboBox.html#firePopupMenuWillBecomeInvisible())()            Notifies PopupMenuListeners that the popup portion of the combo box has become invisible. |
| void | [**firePopupMenuWillBecomeVisible**](http://docs.google.com/javax/swing/JComboBox.html#firePopupMenuWillBecomeVisible())()            Notifies PopupMenuListeners that the popup portion of the combo box will become visible. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JComboBox.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JComboBox. |
| [Action](http://docs.google.com/javax/swing/Action.html) | [**getAction**](http://docs.google.com/javax/swing/JComboBox.html#getAction())()            Returns the currently set Action for this ActionEvent source, or null if no Action is set. |
| [String](http://docs.google.com/java/lang/String.html) | [**getActionCommand**](http://docs.google.com/javax/swing/JComboBox.html#getActionCommand())()            Returns the action command that is included in the event sent to action listeners. |
| [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)[] | [**getActionListeners**](http://docs.google.com/javax/swing/JComboBox.html#getActionListeners())()            Returns an array of all the ActionListeners added to this JComboBox with addActionListener(). |
| [ComboBoxEditor](http://docs.google.com/javax/swing/ComboBoxEditor.html) | [**getEditor**](http://docs.google.com/javax/swing/JComboBox.html#getEditor())()            Returns the editor used to paint and edit the selected item in the JComboBox field. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getItemAt**](http://docs.google.com/javax/swing/JComboBox.html#getItemAt(int))(int index)            Returns the list item at the specified index. |
| int | [**getItemCount**](http://docs.google.com/javax/swing/JComboBox.html#getItemCount())()            Returns the number of items in the list. |
| [ItemListener](http://docs.google.com/java/awt/event/ItemListener.html)[] | [**getItemListeners**](http://docs.google.com/javax/swing/JComboBox.html#getItemListeners())()            Returns an array of all the ItemListeners added to this JComboBox with addItemListener(). |
| [JComboBox.KeySelectionManager](http://docs.google.com/javax/swing/JComboBox.KeySelectionManager.html) | [**getKeySelectionManager**](http://docs.google.com/javax/swing/JComboBox.html#getKeySelectionManager())()            Returns the list's key-selection manager. |
| int | [**getMaximumRowCount**](http://docs.google.com/javax/swing/JComboBox.html#getMaximumRowCount())()            Returns the maximum number of items the combo box can display without a scrollbar |
| [ComboBoxModel](http://docs.google.com/javax/swing/ComboBoxModel.html) | [**getModel**](http://docs.google.com/javax/swing/JComboBox.html#getModel())()            Returns the data model currently used by the JComboBox. |
| [PopupMenuListener](http://docs.google.com/javax/swing/event/PopupMenuListener.html)[] | [**getPopupMenuListeners**](http://docs.google.com/javax/swing/JComboBox.html#getPopupMenuListeners())()            Returns an array of all the PopupMenuListeners added to this JComboBox with addPopupMenuListener(). |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getPrototypeDisplayValue**](http://docs.google.com/javax/swing/JComboBox.html#getPrototypeDisplayValue())()            Returns the "prototypical display" value - an Object used for the calculation of the display height and width. |
| [ListCellRenderer](http://docs.google.com/javax/swing/ListCellRenderer.html) | [**getRenderer**](http://docs.google.com/javax/swing/JComboBox.html#getRenderer())()            Returns the renderer used to display the selected item in the JComboBox field. |
| int | [**getSelectedIndex**](http://docs.google.com/javax/swing/JComboBox.html#getSelectedIndex())()            Returns the first item in the list that matches the given item. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getSelectedItem**](http://docs.google.com/javax/swing/JComboBox.html#getSelectedItem())()            Returns the current selected item. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getSelectedObjects**](http://docs.google.com/javax/swing/JComboBox.html#getSelectedObjects())()            Returns an array containing the selected item. |
| [ComboBoxUI](http://docs.google.com/javax/swing/plaf/ComboBoxUI.html) | [**getUI**](http://docs.google.com/javax/swing/JComboBox.html#getUI())()            Returns the L&F object that renders this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JComboBox.html#getUIClassID())()            Returns the name of the L&F class that renders this component. |
| void | [**hidePopup**](http://docs.google.com/javax/swing/JComboBox.html#hidePopup())()            Causes the combo box to close its popup window. |
| void | [**insertItemAt**](http://docs.google.com/javax/swing/JComboBox.html#insertItemAt(java.lang.Object,%20int))([Object](http://docs.google.com/java/lang/Object.html) anObject, int index)            Inserts an item into the item list at a given index. |
| protected  void | [**installAncestorListener**](http://docs.google.com/javax/swing/JComboBox.html#installAncestorListener())() |
| void | [**intervalAdded**](http://docs.google.com/javax/swing/JComboBox.html#intervalAdded(javax.swing.event.ListDataEvent))([ListDataEvent](http://docs.google.com/javax/swing/event/ListDataEvent.html) e)            This method is public as an implementation side effect. |
| void | [**intervalRemoved**](http://docs.google.com/javax/swing/JComboBox.html#intervalRemoved(javax.swing.event.ListDataEvent))([ListDataEvent](http://docs.google.com/javax/swing/event/ListDataEvent.html) e)            This method is public as an implementation side effect. |
| boolean | [**isEditable**](http://docs.google.com/javax/swing/JComboBox.html#isEditable())()            Returns true if the JComboBox is editable. |
| boolean | [**isLightWeightPopupEnabled**](http://docs.google.com/javax/swing/JComboBox.html#isLightWeightPopupEnabled())()            Gets the value of the lightWeightPopupEnabled property. |
| boolean | [**isPopupVisible**](http://docs.google.com/javax/swing/JComboBox.html#isPopupVisible())()            Determines the visibility of the popup. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JComboBox.html#paramString())()            Returns a string representation of this JComboBox. |
| void | [**processKeyEvent**](http://docs.google.com/javax/swing/JComboBox.html#processKeyEvent(java.awt.event.KeyEvent))([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) e)            Handles KeyEvents, looking for the Tab key. |
| void | [**removeActionListener**](http://docs.google.com/javax/swing/JComboBox.html#removeActionListener(java.awt.event.ActionListener))([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)            Removes an ActionListener. |
| void | [**removeAllItems**](http://docs.google.com/javax/swing/JComboBox.html#removeAllItems())()            Removes all items from the item list. |
| void | [**removeItem**](http://docs.google.com/javax/swing/JComboBox.html#removeItem(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) anObject)            Removes an item from the item list. |
| void | [**removeItemAt**](http://docs.google.com/javax/swing/JComboBox.html#removeItemAt(int))(int anIndex)            Removes the item at anIndex This method works only if the JComboBox uses a mutable data model. |
| void | [**removeItemListener**](http://docs.google.com/javax/swing/JComboBox.html#removeItemListener(java.awt.event.ItemListener))([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) aListener)            Removes an ItemListener. |
| void | [**removePopupMenuListener**](http://docs.google.com/javax/swing/JComboBox.html#removePopupMenuListener(javax.swing.event.PopupMenuListener))([PopupMenuListener](http://docs.google.com/javax/swing/event/PopupMenuListener.html) l)            Removes a PopupMenuListener. |
| protected  void | [**selectedItemChanged**](http://docs.google.com/javax/swing/JComboBox.html#selectedItemChanged())()            This protected method is implementation specific. |
| boolean | [**selectWithKeyChar**](http://docs.google.com/javax/swing/JComboBox.html#selectWithKeyChar(char))(char keyChar)            Selects the list item that corresponds to the specified keyboard character and returns true, if there is an item corresponding to that character. |
| void | [**setAction**](http://docs.google.com/javax/swing/JComboBox.html#setAction(javax.swing.Action))([Action](http://docs.google.com/javax/swing/Action.html) a)            Sets the Action for the ActionEvent source. |
| void | [**setActionCommand**](http://docs.google.com/javax/swing/JComboBox.html#setActionCommand(java.lang.String))([String](http://docs.google.com/java/lang/String.html) aCommand)            Sets the action command that should be included in the event sent to action listeners. |
| void | [**setEditable**](http://docs.google.com/javax/swing/JComboBox.html#setEditable(boolean))(boolean aFlag)            Determines whether the JComboBox field is editable. |
| void | [**setEditor**](http://docs.google.com/javax/swing/JComboBox.html#setEditor(javax.swing.ComboBoxEditor))([ComboBoxEditor](http://docs.google.com/javax/swing/ComboBoxEditor.html) anEditor)            Sets the editor used to paint and edit the selected item in the JComboBox field. |
| void | [**setEnabled**](http://docs.google.com/javax/swing/JComboBox.html#setEnabled(boolean))(boolean b)            Enables the combo box so that items can be selected. |
| void | [**setKeySelectionManager**](http://docs.google.com/javax/swing/JComboBox.html#setKeySelectionManager(javax.swing.JComboBox.KeySelectionManager))([JComboBox.KeySelectionManager](http://docs.google.com/javax/swing/JComboBox.KeySelectionManager.html) aManager)            Sets the object that translates a keyboard character into a list selection. |
| void | [**setLightWeightPopupEnabled**](http://docs.google.com/javax/swing/JComboBox.html#setLightWeightPopupEnabled(boolean))(boolean aFlag)            Sets the lightWeightPopupEnabled property, which provides a hint as to whether or not a lightweight Component should be used to contain the JComboBox, versus a heavyweight Component such as a Panel or a Window. |
| void | [**setMaximumRowCount**](http://docs.google.com/javax/swing/JComboBox.html#setMaximumRowCount(int))(int count)            Sets the maximum number of rows the JComboBox displays. |
| void | [**setModel**](http://docs.google.com/javax/swing/JComboBox.html#setModel(javax.swing.ComboBoxModel))([ComboBoxModel](http://docs.google.com/javax/swing/ComboBoxModel.html) aModel)            Sets the data model that the JComboBox uses to obtain the list of items. |
| void | [**setPopupVisible**](http://docs.google.com/javax/swing/JComboBox.html#setPopupVisible(boolean))(boolean v)            Sets the visibility of the popup. |
| void | [**setPrototypeDisplayValue**](http://docs.google.com/javax/swing/JComboBox.html#setPrototypeDisplayValue(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) prototypeDisplayValue)            Sets the prototype display value used to calculate the size of the display for the UI portion. |
| void | [**setRenderer**](http://docs.google.com/javax/swing/JComboBox.html#setRenderer(javax.swing.ListCellRenderer))([ListCellRenderer](http://docs.google.com/javax/swing/ListCellRenderer.html) aRenderer)            Sets the renderer that paints the list items and the item selected from the list in the JComboBox field. |
| void | [**setSelectedIndex**](http://docs.google.com/javax/swing/JComboBox.html#setSelectedIndex(int))(int anIndex)            Selects the item at index anIndex. |
| void | [**setSelectedItem**](http://docs.google.com/javax/swing/JComboBox.html#setSelectedItem(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) anObject)            Sets the selected item in the combo box display area to the object in the argument. |
| void | [**setUI**](http://docs.google.com/javax/swing/JComboBox.html#setUI(javax.swing.plaf.ComboBoxUI))([ComboBoxUI](http://docs.google.com/javax/swing/plaf/ComboBoxUI.html) ui)            Sets the L&F object that renders this component. |
| void | [**showPopup**](http://docs.google.com/javax/swing/JComboBox.html#showPopup())()            Causes the combo box to display its popup window. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JComboBox.html#updateUI())()            Resets the UI property to a value from the current look and feel. |

| **Methods inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [addAncestorListener](http://docs.google.com/javax/swing/JComponent.html#addAncestorListener(javax.swing.event.AncestorListener)), [addNotify](http://docs.google.com/javax/swing/JComponent.html#addNotify()), [addVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#addVetoableChangeListener(java.beans.VetoableChangeListener)), [computeVisibleRect](http://docs.google.com/javax/swing/JComponent.html#computeVisibleRect(java.awt.Rectangle)), [contains](http://docs.google.com/javax/swing/JComponent.html#contains(int,%20int)), [createToolTip](http://docs.google.com/javax/swing/JComponent.html#createToolTip()), [disable](http://docs.google.com/javax/swing/JComponent.html#disable()), [enable](http://docs.google.com/javax/swing/JComponent.html#enable()), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20int,%20int)), [fireVetoableChange](http://docs.google.com/javax/swing/JComponent.html#fireVetoableChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [getActionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getActionForKeyStroke(javax.swing.KeyStroke)), [getActionMap](http://docs.google.com/javax/swing/JComponent.html#getActionMap()), [getAlignmentX](http://docs.google.com/javax/swing/JComponent.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/javax/swing/JComponent.html#getAlignmentY()), [getAncestorListeners](http://docs.google.com/javax/swing/JComponent.html#getAncestorListeners()), [getAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#getAutoscrolls()), [getBaseline](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/JComponent.html#getBaselineResizeBehavior()), [getBorder](http://docs.google.com/javax/swing/JComponent.html#getBorder()), [getBounds](http://docs.google.com/javax/swing/JComponent.html#getBounds(java.awt.Rectangle)), [getClientProperty](http://docs.google.com/javax/swing/JComponent.html#getClientProperty(java.lang.Object)), [getComponentGraphics](http://docs.google.com/javax/swing/JComponent.html#getComponentGraphics(java.awt.Graphics)), [getComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getComponentPopupMenu()), [getConditionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getConditionForKeyStroke(javax.swing.KeyStroke)), [getDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#getDebugGraphicsOptions()), [getDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale()), [getFontMetrics](http://docs.google.com/javax/swing/JComponent.html#getFontMetrics(java.awt.Font)), [getGraphics](http://docs.google.com/javax/swing/JComponent.html#getGraphics()), [getHeight](http://docs.google.com/javax/swing/JComponent.html#getHeight()), [getInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getInheritsPopupMenu()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap(int)), [getInputVerifier](http://docs.google.com/javax/swing/JComponent.html#getInputVerifier()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets(java.awt.Insets)), [getListeners](http://docs.google.com/javax/swing/JComponent.html#getListeners(java.lang.Class)), [getLocation](http://docs.google.com/javax/swing/JComponent.html#getLocation(java.awt.Point)), [getMaximumSize](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [getNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#getNextFocusableComponent()), [getPopupLocation](http://docs.google.com/javax/swing/JComponent.html#getPopupLocation(java.awt.event.MouseEvent)), [getPreferredSize](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [getRegisteredKeyStrokes](http://docs.google.com/javax/swing/JComponent.html#getRegisteredKeyStrokes()), [getRootPane](http://docs.google.com/javax/swing/JComponent.html#getRootPane()), [getSize](http://docs.google.com/javax/swing/JComponent.html#getSize(java.awt.Dimension)), [getToolTipLocation](http://docs.google.com/javax/swing/JComponent.html#getToolTipLocation(java.awt.event.MouseEvent)), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText()), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText(java.awt.event.MouseEvent)), [getTopLevelAncestor](http://docs.google.com/javax/swing/JComponent.html#getTopLevelAncestor()), [getTransferHandler](http://docs.google.com/javax/swing/JComponent.html#getTransferHandler()), [getVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#getVerifyInputWhenFocusTarget()), [getVetoableChangeListeners](http://docs.google.com/javax/swing/JComponent.html#getVetoableChangeListeners()), [getVisibleRect](http://docs.google.com/javax/swing/JComponent.html#getVisibleRect()), [getWidth](http://docs.google.com/javax/swing/JComponent.html#getWidth()), [getX](http://docs.google.com/javax/swing/JComponent.html#getX()), [getY](http://docs.google.com/javax/swing/JComponent.html#getY()), [grabFocus](http://docs.google.com/javax/swing/JComponent.html#grabFocus()), [isDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#isDoubleBuffered()), [isLightweightComponent](http://docs.google.com/javax/swing/JComponent.html#isLightweightComponent(java.awt.Component)), [isManagingFocus](http://docs.google.com/javax/swing/JComponent.html#isManagingFocus()), [isOpaque](http://docs.google.com/javax/swing/JComponent.html#isOpaque()), [isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()), [isPaintingForPrint](http://docs.google.com/javax/swing/JComponent.html#isPaintingForPrint()), [isPaintingTile](http://docs.google.com/javax/swing/JComponent.html#isPaintingTile()), [isRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#isRequestFocusEnabled()), [isValidateRoot](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot()), [paint](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [paintBorder](http://docs.google.com/javax/swing/JComponent.html#paintBorder(java.awt.Graphics)), [paintChildren](http://docs.google.com/javax/swing/JComponent.html#paintChildren(java.awt.Graphics)), [paintComponent](http://docs.google.com/javax/swing/JComponent.html#paintComponent(java.awt.Graphics)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(int,%20int,%20int,%20int)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(java.awt.Rectangle)), [print](http://docs.google.com/javax/swing/JComponent.html#print(java.awt.Graphics)), [printAll](http://docs.google.com/javax/swing/JComponent.html#printAll(java.awt.Graphics)), [printBorder](http://docs.google.com/javax/swing/JComponent.html#printBorder(java.awt.Graphics)), [printChildren](http://docs.google.com/javax/swing/JComponent.html#printChildren(java.awt.Graphics)), [printComponent](http://docs.google.com/javax/swing/JComponent.html#printComponent(java.awt.Graphics)), [processComponentKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processComponentKeyEvent(java.awt.event.KeyEvent)), [processKeyBinding](http://docs.google.com/javax/swing/JComponent.html#processKeyBinding(javax.swing.KeyStroke,%20java.awt.event.KeyEvent,%20int,%20boolean)), [processMouseEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [putClientProperty](http://docs.google.com/javax/swing/JComponent.html#putClientProperty(java.lang.Object,%20java.lang.Object)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20javax.swing.KeyStroke,%20int)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20java.lang.String,%20javax.swing.KeyStroke,%20int)), [removeAncestorListener](http://docs.google.com/javax/swing/JComponent.html#removeAncestorListener(javax.swing.event.AncestorListener)), [removeNotify](http://docs.google.com/javax/swing/JComponent.html#removeNotify()), [removeVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#removeVetoableChangeListener(java.beans.VetoableChangeListener)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(java.awt.Rectangle)), [requestDefaultFocus](http://docs.google.com/javax/swing/JComponent.html#requestDefaultFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow(boolean)), [resetKeyboardActions](http://docs.google.com/javax/swing/JComponent.html#resetKeyboardActions()), [reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)), [revalidate](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [scrollRectToVisible](http://docs.google.com/javax/swing/JComponent.html#scrollRectToVisible(java.awt.Rectangle)), [setActionMap](http://docs.google.com/javax/swing/JComponent.html#setActionMap(javax.swing.ActionMap)), [setAlignmentX](http://docs.google.com/javax/swing/JComponent.html#setAlignmentX(float)), [setAlignmentY](http://docs.google.com/javax/swing/JComponent.html#setAlignmentY(float)), [setAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#setAutoscrolls(boolean)), [setBackground](http://docs.google.com/javax/swing/JComponent.html#setBackground(java.awt.Color)), [setBorder](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)), [setComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setComponentPopupMenu(javax.swing.JPopupMenu)), [setDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#setDebugGraphicsOptions(int)), [setDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#setDefaultLocale(java.util.Locale)), [setDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#setDoubleBuffered(boolean)), [setFocusTraversalKeys](http://docs.google.com/javax/swing/JComponent.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFont](http://docs.google.com/javax/swing/JComponent.html#setFont(java.awt.Font)), [setForeground](http://docs.google.com/javax/swing/JComponent.html#setForeground(java.awt.Color)), [setInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setInheritsPopupMenu(boolean)), [setInputMap](http://docs.google.com/javax/swing/JComponent.html#setInputMap(int,%20javax.swing.InputMap)), [setInputVerifier](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [setMaximumSize](http://docs.google.com/javax/swing/JComponent.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/javax/swing/JComponent.html#setMinimumSize(java.awt.Dimension)), [setNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#setNextFocusableComponent(java.awt.Component)), [setOpaque](http://docs.google.com/javax/swing/JComponent.html#setOpaque(boolean)), [setPreferredSize](http://docs.google.com/javax/swing/JComponent.html#setPreferredSize(java.awt.Dimension)), [setRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#setRequestFocusEnabled(boolean)), [setToolTipText](http://docs.google.com/javax/swing/JComponent.html#setToolTipText(java.lang.String)), [setTransferHandler](http://docs.google.com/javax/swing/JComponent.html#setTransferHandler(javax.swing.TransferHandler)), [setUI](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [setVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#setVerifyInputWhenFocusTarget(boolean)), [setVisible](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [unregisterKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#unregisterKeyboardAction(javax.swing.KeyStroke)), [update](http://docs.google.com/javax/swing/JComponent.html#update(java.awt.Graphics)) |

| **Methods inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.lang.String,%20java.awt.Component)), [addContainerListener](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener)), [addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [applyComponentOrientation](http://docs.google.com/java/awt/Container.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Container.html#areFocusTraversalKeysSet(int)), [countComponents](http://docs.google.com/java/awt/Container.html#countComponents()), [deliverEvent](http://docs.google.com/java/awt/Container.html#deliverEvent(java.awt.Event)), [doLayout](http://docs.google.com/java/awt/Container.html#doLayout()), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(int,%20int)), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(java.awt.Point)), [getComponent](http://docs.google.com/java/awt/Container.html#getComponent(int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(int,%20int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(java.awt.Point)), [getComponentCount](http://docs.google.com/java/awt/Container.html#getComponentCount()), [getComponents](http://docs.google.com/java/awt/Container.html#getComponents()), [getComponentZOrder](http://docs.google.com/java/awt/Container.html#getComponentZOrder(java.awt.Component)), [getContainerListeners](http://docs.google.com/java/awt/Container.html#getContainerListeners()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Container.html#getFocusTraversalKeys(int)), [getFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [getLayout](http://docs.google.com/java/awt/Container.html#getLayout()), [getMousePosition](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean)), [insets](http://docs.google.com/java/awt/Container.html#insets()), [invalidate](http://docs.google.com/java/awt/Container.html#invalidate()), [isAncestorOf](http://docs.google.com/java/awt/Container.html#isAncestorOf(java.awt.Component)), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot()), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container)), [isFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicyProvider()), [isFocusTraversalPolicySet](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicySet()), [layout](http://docs.google.com/java/awt/Container.html#layout()), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintStream,%20int)), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintWriter,%20int)), [locate](http://docs.google.com/java/awt/Container.html#locate(int,%20int)), [minimumSize](http://docs.google.com/java/awt/Container.html#minimumSize()), [paintComponents](http://docs.google.com/java/awt/Container.html#paintComponents(java.awt.Graphics)), [preferredSize](http://docs.google.com/java/awt/Container.html#preferredSize()), [printComponents](http://docs.google.com/java/awt/Container.html#printComponents(java.awt.Graphics)), [processContainerEvent](http://docs.google.com/java/awt/Container.html#processContainerEvent(java.awt.event.ContainerEvent)), [processEvent](http://docs.google.com/java/awt/Container.html#processEvent(java.awt.AWTEvent)), [remove](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)), [remove](http://docs.google.com/java/awt/Container.html#remove(int)), [removeAll](http://docs.google.com/java/awt/Container.html#removeAll()), [removeContainerListener](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener)), [setComponentZOrder](http://docs.google.com/java/awt/Container.html#setComponentZOrder(java.awt.Component,%20int)), [setFocusCycleRoot](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [validate](http://docs.google.com/java/awt/Container.html#validate()), [validateTree](http://docs.google.com/java/awt/Container.html#validateTree()) |

| **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(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)), [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(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,%20byte,%20byte)), [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,%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)), [getBackground](http://docs.google.com/java/awt/Component.html#getBackground()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [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()), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [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()), [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()), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [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()), [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()), [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)), [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)), [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)), [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)), [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)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [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)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [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)), [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)) |

| **Field Detail** |
| --- |

### dataModel

protected [ComboBoxModel](http://docs.google.com/javax/swing/ComboBoxModel.html) **dataModel**

This protected field is implementation specific. Do not access directly or override. Use the accessor methods instead.

**See Also:**[getModel()](http://docs.google.com/javax/swing/JComboBox.html#getModel()), [setModel(javax.swing.ComboBoxModel)](http://docs.google.com/javax/swing/JComboBox.html#setModel(javax.swing.ComboBoxModel))

### renderer

protected [ListCellRenderer](http://docs.google.com/javax/swing/ListCellRenderer.html) **renderer**

This protected field is implementation specific. Do not access directly or override. Use the accessor methods instead.

**See Also:**[getRenderer()](http://docs.google.com/javax/swing/JComboBox.html#getRenderer()), [setRenderer(javax.swing.ListCellRenderer)](http://docs.google.com/javax/swing/JComboBox.html#setRenderer(javax.swing.ListCellRenderer))

### editor

protected [ComboBoxEditor](http://docs.google.com/javax/swing/ComboBoxEditor.html) **editor**

This protected field is implementation specific. Do not access directly or override. Use the accessor methods instead.

**See Also:**[getEditor()](http://docs.google.com/javax/swing/JComboBox.html#getEditor()), [setEditor(javax.swing.ComboBoxEditor)](http://docs.google.com/javax/swing/JComboBox.html#setEditor(javax.swing.ComboBoxEditor))

### maximumRowCount

protected int **maximumRowCount**

This protected field is implementation specific. Do not access directly or override. Use the accessor methods instead.

**See Also:**[getMaximumRowCount()](http://docs.google.com/javax/swing/JComboBox.html#getMaximumRowCount()), [setMaximumRowCount(int)](http://docs.google.com/javax/swing/JComboBox.html#setMaximumRowCount(int))

### isEditable

protected boolean **isEditable**

This protected field is implementation specific. Do not access directly or override. Use the accessor methods instead.

**See Also:**[isEditable](http://docs.google.com/javax/swing/JComboBox.html#isEditable), [setEditable(boolean)](http://docs.google.com/javax/swing/JComboBox.html#setEditable(boolean))

### keySelectionManager

protected [JComboBox.KeySelectionManager](http://docs.google.com/javax/swing/JComboBox.KeySelectionManager.html) **keySelectionManager**

This protected field is implementation specific. Do not access directly or override. Use the accessor methods instead.

**See Also:**[setKeySelectionManager(javax.swing.JComboBox.KeySelectionManager)](http://docs.google.com/javax/swing/JComboBox.html#setKeySelectionManager(javax.swing.JComboBox.KeySelectionManager)), [getKeySelectionManager()](http://docs.google.com/javax/swing/JComboBox.html#getKeySelectionManager())

### actionCommand

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

This protected field is implementation specific. Do not access directly or override. Use the accessor methods instead.

**See Also:**[setActionCommand(java.lang.String)](http://docs.google.com/javax/swing/JComboBox.html#setActionCommand(java.lang.String)), [getActionCommand()](http://docs.google.com/javax/swing/JComboBox.html#getActionCommand())

### lightWeightPopupEnabled

protected boolean **lightWeightPopupEnabled**

This protected field is implementation specific. Do not access directly or override. Use the accessor methods instead.

**See Also:**[setLightWeightPopupEnabled(boolean)](http://docs.google.com/javax/swing/JComboBox.html#setLightWeightPopupEnabled(boolean)), [isLightWeightPopupEnabled()](http://docs.google.com/javax/swing/JComboBox.html#isLightWeightPopupEnabled())

### selectedItemReminder

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

This protected field is implementation specific. Do not access directly or override.

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

### JComboBox

public **JComboBox**([ComboBoxModel](http://docs.google.com/javax/swing/ComboBoxModel.html) aModel)

Creates a JComboBox that takes its items from an existing ComboBoxModel. Since the ComboBoxModel is provided, a combo box created using this constructor does not create a default combo box model and may impact how the insert, remove and add methods behave.

**Parameters:**aModel - the ComboBoxModel that provides the displayed list of items**See Also:**[DefaultComboBoxModel](http://docs.google.com/javax/swing/DefaultComboBoxModel.html)

### JComboBox

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

Creates a JComboBox that contains the elements in the specified array. By default the first item in the array (and therefore the data model) becomes selected.

**Parameters:**items - an array of objects to insert into the combo box**See Also:**[DefaultComboBoxModel](http://docs.google.com/javax/swing/DefaultComboBoxModel.html)

### JComboBox

public **JComboBox**([Vector](http://docs.google.com/java/util/Vector.html)<?> items)

Creates a JComboBox that contains the elements in the specified Vector. By default the first item in the vector (and therefore the data model) becomes selected.

**Parameters:**items - an array of vectors to insert into the combo box**See Also:**[DefaultComboBoxModel](http://docs.google.com/javax/swing/DefaultComboBoxModel.html)

### JComboBox

public **JComboBox**()

Creates a JComboBox with a default data model. The default data model is an empty list of objects. Use addItem to add items. By default the first item in the data model becomes selected.

**See Also:**[DefaultComboBoxModel](http://docs.google.com/javax/swing/DefaultComboBoxModel.html)

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

### installAncestorListener

protected void **installAncestorListener**()

### setUI

public void **setUI**([ComboBoxUI](http://docs.google.com/javax/swing/plaf/ComboBoxUI.html) ui)

Sets the L&F object that renders this component.

**Parameters:**ui - the ComboBoxUI L&F object**See Also:**[UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### updateUI

public void **updateUI**()

Resets the UI property to a value from the current look and feel.

**Overrides:**[updateUI](http://docs.google.com/javax/swing/JComponent.html#updateUI()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **See Also:**[JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### getUIClassID

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

Returns the name of the L&F class that renders this component.

**Overrides:**[getUIClassID](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the string "ComboBoxUI"**See Also:**[JComponent.getUIClassID()](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()), [UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### getUI

public [ComboBoxUI](http://docs.google.com/javax/swing/plaf/ComboBoxUI.html) **getUI**()

Returns the L&F object that renders this component.

**Returns:**the ComboBoxUI object that renders this component

### setModel

public void **setModel**([ComboBoxModel](http://docs.google.com/javax/swing/ComboBoxModel.html) aModel)

Sets the data model that the JComboBox uses to obtain the list of items.

**Parameters:**aModel - the ComboBoxModel that provides the displayed list of items

### getModel

public [ComboBoxModel](http://docs.google.com/javax/swing/ComboBoxModel.html) **getModel**()

Returns the data model currently used by the JComboBox.

**Returns:**the ComboBoxModel that provides the displayed list of items

### setLightWeightPopupEnabled

public void **setLightWeightPopupEnabled**(boolean aFlag)

Sets the lightWeightPopupEnabled property, which provides a hint as to whether or not a lightweight Component should be used to contain the JComboBox, versus a heavyweight Component such as a Panel or a Window. The decision of lightweight versus heavyweight is ultimately up to the JComboBox. Lightweight windows are more efficient than heavyweight windows, but lightweight and heavyweight components do not mix well in a GUI. If your application mixes lightweight and heavyweight components, you should disable lightweight popups. The default value for the lightWeightPopupEnabled property is true, unless otherwise specified by the look and feel. Some look and feels always use heavyweight popups, no matter what the value of this property.

See the article [Mixing Heavy and Light Components](http://java.sun.com/products/jfc/tsc/articles/mixing/index.html) on  [*The Swing Connection*](http://java.sun.com/products/jfc/tsc) This method fires a property changed event.

**Parameters:**aFlag - if true, lightweight popups are desired

### isLightWeightPopupEnabled

public boolean **isLightWeightPopupEnabled**()

Gets the value of the lightWeightPopupEnabled property.

**Returns:**the value of the lightWeightPopupEnabled property**See Also:**[setLightWeightPopupEnabled(boolean)](http://docs.google.com/javax/swing/JComboBox.html#setLightWeightPopupEnabled(boolean))

### setEditable

public void **setEditable**(boolean aFlag)

Determines whether the JComboBox field is editable. An editable JComboBox allows the user to type into the field or selected an item from the list to initialize the field, after which it can be edited. (The editing affects only the field, the list item remains intact.) A non editable JComboBox displays the selected item in the field, but the selection cannot be modified.

**Parameters:**aFlag - a boolean value, where true indicates that the field is editable

### isEditable

public boolean **isEditable**()

Returns true if the JComboBox is editable. By default, a combo box is not editable.

**Returns:**true if the JComboBox is editable, else false

### setMaximumRowCount

public void **setMaximumRowCount**(int count)

Sets the maximum number of rows the JComboBox displays. If the number of objects in the model is greater than count, the combo box uses a scrollbar.

**Parameters:**count - an integer specifying the maximum number of items to display in the list before using a scrollbar

### getMaximumRowCount

public int **getMaximumRowCount**()

Returns the maximum number of items the combo box can display without a scrollbar

**Returns:**an integer specifying the maximum number of items that are displayed in the list before using a scrollbar

### setRenderer

public void **setRenderer**([ListCellRenderer](http://docs.google.com/javax/swing/ListCellRenderer.html) aRenderer)

Sets the renderer that paints the list items and the item selected from the list in the JComboBox field. The renderer is used if the JComboBox is not editable. If it is editable, the editor is used to render and edit the selected item.

The default renderer displays a string or an icon. Other renderers can handle graphic images and composite items.

To display the selected item, aRenderer.getListCellRendererComponent is called, passing the list object and an index of -1.

**Parameters:**aRenderer - the ListCellRenderer that displays the selected item**See Also:**[setEditor(javax.swing.ComboBoxEditor)](http://docs.google.com/javax/swing/JComboBox.html#setEditor(javax.swing.ComboBoxEditor))

### getRenderer

public [ListCellRenderer](http://docs.google.com/javax/swing/ListCellRenderer.html) **getRenderer**()

Returns the renderer used to display the selected item in the JComboBox field.

**Returns:**the ListCellRenderer that displays the selected item.

### setEditor

public void **setEditor**([ComboBoxEditor](http://docs.google.com/javax/swing/ComboBoxEditor.html) anEditor)

Sets the editor used to paint and edit the selected item in the JComboBox field. The editor is used only if the receiving JComboBox is editable. If not editable, the combo box uses the renderer to paint the selected item.

**Parameters:**anEditor - the ComboBoxEditor that displays the selected item**See Also:**[setRenderer(javax.swing.ListCellRenderer)](http://docs.google.com/javax/swing/JComboBox.html#setRenderer(javax.swing.ListCellRenderer))

### getEditor

public [ComboBoxEditor](http://docs.google.com/javax/swing/ComboBoxEditor.html) **getEditor**()

Returns the editor used to paint and edit the selected item in the JComboBox field.

**Returns:**the ComboBoxEditor that displays the selected item

### setSelectedItem

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

Sets the selected item in the combo box display area to the object in the argument. If anObject is in the list, the display area shows anObject selected.

If anObject is *not* in the list and the combo box is uneditable, it will not change the current selection. For editable combo boxes, the selection will change to anObject.

If this constitutes a change in the selected item, ItemListeners added to the combo box will be notified with one or two ItemEvents. If there is a current selected item, an ItemEvent will be fired and the state change will be ItemEvent.DESELECTED. If anObject is in the list and is not currently selected then an ItemEvent will be fired and the state change will be ItemEvent.SELECTED.

ActionListeners added to the combo box will be notified with an ActionEvent when this method is called.

**Parameters:**anObject - the list object to select; use null to clear the selection

### getSelectedItem

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

Returns the current selected item.

If the combo box is editable, then this value may not have been added to the combo box with addItem, insertItemAt or the data constructors.

**Returns:**the current selected Object**See Also:**[setSelectedItem(java.lang.Object)](http://docs.google.com/javax/swing/JComboBox.html#setSelectedItem(java.lang.Object))

### setSelectedIndex

public void **setSelectedIndex**(int anIndex)

Selects the item at index anIndex.

**Parameters:**anIndex - an integer specifying the list item to select, where 0 specifies the first item in the list and -1 indicates no selection **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if anIndex < -1 or anIndex is greater than or equal to size

### getSelectedIndex

public int **getSelectedIndex**()

Returns the first item in the list that matches the given item. The result is not always defined if the JComboBox allows selected items that are not in the list. Returns -1 if there is no selected item or if the user specified an item which is not in the list.

**Returns:**an integer specifying the currently selected list item, where 0 specifies the first item in the list; or -1 if no item is selected or if the currently selected item is not in the list

### getPrototypeDisplayValue

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

Returns the "prototypical display" value - an Object used for the calculation of the display height and width.

**Returns:**the value of the prototypeDisplayValue property**Since:** 1.4 **See Also:**[setPrototypeDisplayValue(java.lang.Object)](http://docs.google.com/javax/swing/JComboBox.html#setPrototypeDisplayValue(java.lang.Object))

### setPrototypeDisplayValue

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

Sets the prototype display value used to calculate the size of the display for the UI portion.

If a prototype display value is specified, the preferred size of the combo box is calculated by configuring the renderer with the prototype display value and obtaining its preferred size. Specifying the preferred display value is often useful when the combo box will be displaying large amounts of data. If no prototype display value has been specified, the renderer must be configured for each value from the model and its preferred size obtained, which can be relatively expensive.

**Parameters:**prototypeDisplayValue - **Since:** 1.4 **See Also:**[getPrototypeDisplayValue()](http://docs.google.com/javax/swing/JComboBox.html#getPrototypeDisplayValue())

### addItem

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

Adds an item to the item list. This method works only if the JComboBox uses a mutable data model.

**Warning:** Focus and keyboard navigation problems may arise if you add duplicate String objects. A workaround is to add new objects instead of String objects and make sure that the toString() method is defined. For example:

comboBox.addItem(makeObj("Item 1"));  
 comboBox.addItem(makeObj("Item 1"));  
 ...  
 private Object makeObj(final String item) {  
 return new Object() { public String toString() { return item; } };  
 }

**Parameters:**anObject - the Object to add to the list**See Also:**[MutableComboBoxModel](http://docs.google.com/javax/swing/MutableComboBoxModel.html)

### insertItemAt

public void **insertItemAt**([Object](http://docs.google.com/java/lang/Object.html) anObject,  
 int index)

Inserts an item into the item list at a given index. This method works only if the JComboBox uses a mutable data model.

**Parameters:**anObject - the Object to add to the listindex - an integer specifying the position at which to add the item**See Also:**[MutableComboBoxModel](http://docs.google.com/javax/swing/MutableComboBoxModel.html)

### removeItem

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

Removes an item from the item list. This method works only if the JComboBox uses a mutable data model.

**Parameters:**anObject - the object to remove from the item list**See Also:**[MutableComboBoxModel](http://docs.google.com/javax/swing/MutableComboBoxModel.html)

### removeItemAt

public void **removeItemAt**(int anIndex)

Removes the item at anIndex This method works only if the JComboBox uses a mutable data model.

**Parameters:**anIndex - an int specifying the index of the item to remove, where 0 indicates the first item in the list**See Also:**[MutableComboBoxModel](http://docs.google.com/javax/swing/MutableComboBoxModel.html)

### removeAllItems

public void **removeAllItems**()

Removes all items from the item list.

### showPopup

public void **showPopup**()

Causes the combo box to display its popup window.

**See Also:**[setPopupVisible(boolean)](http://docs.google.com/javax/swing/JComboBox.html#setPopupVisible(boolean))

### hidePopup

public void **hidePopup**()

Causes the combo box to close its popup window.

**See Also:**[setPopupVisible(boolean)](http://docs.google.com/javax/swing/JComboBox.html#setPopupVisible(boolean))

### setPopupVisible

public void **setPopupVisible**(boolean v)

Sets the visibility of the popup.

### isPopupVisible

public boolean **isPopupVisible**()

Determines the visibility of the popup.

**Returns:**true if the popup is visible, otherwise returns false

### addItemListener

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

Adds an ItemListener.

aListener will receive one or two ItemEvents when the selected item changes.

**Specified by:**[addItemListener](http://docs.google.com/java/awt/ItemSelectable.html#addItemListener(java.awt.event.ItemListener)) in interface [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html) **Parameters:**aListener - the ItemListener that is to be notified**See Also:**[setSelectedItem(java.lang.Object)](http://docs.google.com/javax/swing/JComboBox.html#setSelectedItem(java.lang.Object))

### removeItemListener

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

Removes an ItemListener.

**Specified by:**[removeItemListener](http://docs.google.com/java/awt/ItemSelectable.html#removeItemListener(java.awt.event.ItemListener)) in interface [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html) **Parameters:**aListener - the ItemListener to remove**See Also:**[ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html)

### getItemListeners

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

Returns an array of all the ItemListeners added to this JComboBox with addItemListener().

**Returns:**all of the ItemListeners added or an empty array if no listeners have been added**Since:** 1.4

### addActionListener

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

Adds an ActionListener.

The ActionListener will receive an ActionEvent when a selection has been made. If the combo box is editable, then an ActionEvent will be fired when editing has stopped.

**Parameters:**l - the ActionListener that is to be notified**See Also:**[setSelectedItem(java.lang.Object)](http://docs.google.com/javax/swing/JComboBox.html#setSelectedItem(java.lang.Object))

### removeActionListener

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

Removes an ActionListener.

**Parameters:**l - the ActionListener to remove

### getActionListeners

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

Returns an array of all the ActionListeners added to this JComboBox with addActionListener().

**Returns:**all of the ActionListeners added or an empty array if no listeners have been added**Since:** 1.4

### addPopupMenuListener

public void **addPopupMenuListener**([PopupMenuListener](http://docs.google.com/javax/swing/event/PopupMenuListener.html) l)

Adds a PopupMenu listener which will listen to notification messages from the popup portion of the combo box.

For all standard look and feels shipped with Java, the popup list portion of combo box is implemented as a JPopupMenu. A custom look and feel may not implement it this way and will therefore not receive the notification.

**Parameters:**l - the PopupMenuListener to add**Since:** 1.4

### removePopupMenuListener

public void **removePopupMenuListener**([PopupMenuListener](http://docs.google.com/javax/swing/event/PopupMenuListener.html) l)

Removes a PopupMenuListener.

**Parameters:**l - the PopupMenuListener to remove**Since:** 1.4 **See Also:**[addPopupMenuListener(javax.swing.event.PopupMenuListener)](http://docs.google.com/javax/swing/JComboBox.html#addPopupMenuListener(javax.swing.event.PopupMenuListener))

### getPopupMenuListeners

public [PopupMenuListener](http://docs.google.com/javax/swing/event/PopupMenuListener.html)[] **getPopupMenuListeners**()

Returns an array of all the PopupMenuListeners added to this JComboBox with addPopupMenuListener().

**Returns:**all of the PopupMenuListeners added or an empty array if no listeners have been added**Since:** 1.4

### firePopupMenuWillBecomeVisible

public void **firePopupMenuWillBecomeVisible**()

Notifies PopupMenuListeners that the popup portion of the combo box will become visible.

This method is public but should not be called by anything other than the UI delegate.

**Since:** 1.4 **See Also:**[addPopupMenuListener(javax.swing.event.PopupMenuListener)](http://docs.google.com/javax/swing/JComboBox.html#addPopupMenuListener(javax.swing.event.PopupMenuListener))

### firePopupMenuWillBecomeInvisible

public void **firePopupMenuWillBecomeInvisible**()

Notifies PopupMenuListeners that the popup portion of the combo box has become invisible.

This method is public but should not be called by anything other than the UI delegate.

**Since:** 1.4 **See Also:**[addPopupMenuListener(javax.swing.event.PopupMenuListener)](http://docs.google.com/javax/swing/JComboBox.html#addPopupMenuListener(javax.swing.event.PopupMenuListener))

### firePopupMenuCanceled

public void **firePopupMenuCanceled**()

Notifies PopupMenuListeners that the popup portion of the combo box has been canceled.

This method is public but should not be called by anything other than the UI delegate.

**Since:** 1.4 **See Also:**[addPopupMenuListener(javax.swing.event.PopupMenuListener)](http://docs.google.com/javax/swing/JComboBox.html#addPopupMenuListener(javax.swing.event.PopupMenuListener))

### setActionCommand

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

Sets the action command that should be included in the event sent to action listeners.

**Parameters:**aCommand - a string containing the "command" that is sent to action listeners; the same listener can then do different things depending on the command it receives

### getActionCommand

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

Returns the action command that is included in the event sent to action listeners.

**Returns:**the string containing the "command" that is sent to action listeners.

### setAction

public void **setAction**([Action](http://docs.google.com/javax/swing/Action.html) a)

Sets the Action for the ActionEvent source. The new Action replaces any previously set Action but does not affect ActionListeners independently added with addActionListener. If the Action is already a registered ActionListener for the ActionEvent source, it is not re-registered.

Setting the Action results in immediately changing all the properties described in  [Swing Components Supporting Action](http://docs.google.com/Action.html#buttonActions). Subsequently, the combobox's properties are automatically updated as the Action's properties change.

This method uses three other methods to set and help track the Action's property values. It uses the configurePropertiesFromAction method to immediately change the combobox's properties. To track changes in the Action's property values, this method registers the PropertyChangeListener returned by createActionPropertyChangeListener. The default PropertyChangeListener invokes the actionPropertyChanged method when a property in the Action changes.

**Parameters:**a - the Action for the JComboBox, or null.**Since:** 1.3 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [getAction()](http://docs.google.com/javax/swing/JComboBox.html#getAction()), [configurePropertiesFromAction(javax.swing.Action)](http://docs.google.com/javax/swing/JComboBox.html#configurePropertiesFromAction(javax.swing.Action)), [createActionPropertyChangeListener(javax.swing.Action)](http://docs.google.com/javax/swing/JComboBox.html#createActionPropertyChangeListener(javax.swing.Action)), [actionPropertyChanged(javax.swing.Action, java.lang.String)](http://docs.google.com/javax/swing/JComboBox.html#actionPropertyChanged(javax.swing.Action,%20java.lang.String))

### getAction

public [Action](http://docs.google.com/javax/swing/Action.html) **getAction**()

Returns the currently set Action for this ActionEvent source, or null if no Action is set.

**Returns:**the Action for this ActionEvent source; or null**Since:** 1.3 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [setAction(javax.swing.Action)](http://docs.google.com/javax/swing/JComboBox.html#setAction(javax.swing.Action))

### configurePropertiesFromAction

protected void **configurePropertiesFromAction**([Action](http://docs.google.com/javax/swing/Action.html) a)

Sets the properties on this combobox to match those in the specified Action. Refer to  [Swing Components Supporting Action](http://docs.google.com/Action.html#buttonActions) for more details as to which properties this sets.

**Parameters:**a - the Action from which to get the properties, or null**Since:** 1.3 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [setAction(javax.swing.Action)](http://docs.google.com/javax/swing/JComboBox.html#setAction(javax.swing.Action))

### createActionPropertyChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **createActionPropertyChangeListener**([Action](http://docs.google.com/javax/swing/Action.html) a)

Creates and returns a PropertyChangeListener that is responsible for listening for changes from the specified Action and updating the appropriate properties.

**Warning:** If you subclass this do not create an anonymous inner class. If you do the lifetime of the combobox will be tied to that of the Action.

**Parameters:**a - the combobox's action**Since:** 1.3 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [setAction(javax.swing.Action)](http://docs.google.com/javax/swing/JComboBox.html#setAction(javax.swing.Action))

### actionPropertyChanged

protected void **actionPropertyChanged**([Action](http://docs.google.com/javax/swing/Action.html) action,  
 [String](http://docs.google.com/java/lang/String.html) propertyName)

Updates the combobox's state in response to property changes in associated action. This method is invoked from the PropertyChangeListener returned from createActionPropertyChangeListener. Subclasses do not normally need to invoke this. Subclasses that support additional Action properties should override this and configurePropertiesFromAction.

Refer to the table at  [Swing Components Supporting Action](http://docs.google.com/Action.html#buttonActions) for a list of the properties this method sets.

**Parameters:**action - the Action associated with this comboboxpropertyName - the name of the property that changed**Since:** 1.6 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [configurePropertiesFromAction(javax.swing.Action)](http://docs.google.com/javax/swing/JComboBox.html#configurePropertiesFromAction(javax.swing.Action))

### fireItemStateChanged

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

Notifies all listeners that have registered interest for notification on this event type.

**Parameters:**e - the event of interest**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### fireActionEvent

protected void **fireActionEvent**()

Notifies all listeners that have registered interest for notification on this event type.

**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### selectedItemChanged

protected void **selectedItemChanged**()

This protected method is implementation specific. Do not access directly or override.

### getSelectedObjects

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

Returns an array containing the selected item. This method is implemented for compatibility with ItemSelectable.

**Specified by:**[getSelectedObjects](http://docs.google.com/java/awt/ItemSelectable.html#getSelectedObjects()) in interface [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html) **Returns:**an array of Objects containing one element -- the selected item

### actionPerformed

public void **actionPerformed**([ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html) e)

This method is public as an implementation side effect. do not call or override.

**Specified by:**[actionPerformed](http://docs.google.com/java/awt/event/ActionListener.html#actionPerformed(java.awt.event.ActionEvent)) in interface [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)

### contentsChanged

public void **contentsChanged**([ListDataEvent](http://docs.google.com/javax/swing/event/ListDataEvent.html) e)

This method is public as an implementation side effect. do not call or override.

**Specified by:**[contentsChanged](http://docs.google.com/javax/swing/event/ListDataListener.html#contentsChanged(javax.swing.event.ListDataEvent)) in interface [ListDataListener](http://docs.google.com/javax/swing/event/ListDataListener.html) **Parameters:**e - a ListDataEvent encapsulating the event information

### intervalAdded

public void **intervalAdded**([ListDataEvent](http://docs.google.com/javax/swing/event/ListDataEvent.html) e)

This method is public as an implementation side effect. do not call or override.

**Specified by:**[intervalAdded](http://docs.google.com/javax/swing/event/ListDataListener.html#intervalAdded(javax.swing.event.ListDataEvent)) in interface [ListDataListener](http://docs.google.com/javax/swing/event/ListDataListener.html) **Parameters:**e - a ListDataEvent encapsulating the event information

### intervalRemoved

public void **intervalRemoved**([ListDataEvent](http://docs.google.com/javax/swing/event/ListDataEvent.html) e)

This method is public as an implementation side effect. do not call or override.

**Specified by:**[intervalRemoved](http://docs.google.com/javax/swing/event/ListDataListener.html#intervalRemoved(javax.swing.event.ListDataEvent)) in interface [ListDataListener](http://docs.google.com/javax/swing/event/ListDataListener.html) **Parameters:**e - a ListDataEvent encapsulating the event information

### selectWithKeyChar

public boolean **selectWithKeyChar**(char keyChar)

Selects the list item that corresponds to the specified keyboard character and returns true, if there is an item corresponding to that character. Otherwise, returns false.

**Parameters:**keyChar - a char, typically this is a keyboard key typed by the user

### setEnabled

public void **setEnabled**(boolean b)

Enables the combo box so that items can be selected. When the combo box is disabled, items cannot be selected and values cannot be typed into its field (if it is editable).

**Overrides:**[setEnabled](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**b - a boolean value, where true enables the component and false disables it**See Also:**[Component.isEnabled()](http://docs.google.com/java/awt/Component.html#isEnabled()), [Component.isLightweight()](http://docs.google.com/java/awt/Component.html#isLightweight())

### configureEditor

public void **configureEditor**([ComboBoxEditor](http://docs.google.com/javax/swing/ComboBoxEditor.html) anEditor,  
 [Object](http://docs.google.com/java/lang/Object.html) anItem)

Initializes the editor with the specified item.

**Parameters:**anEditor - the ComboBoxEditor that displays the list item in the combo box field and allows it to be editedanItem - the object to display and edit in the field

### processKeyEvent

public void **processKeyEvent**([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) e)

Handles KeyEvents, looking for the Tab key. If the Tab key is found, the popup window is closed.

**Overrides:**[processKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processKeyEvent(java.awt.event.KeyEvent)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**e - the KeyEvent containing the keyboard key that was pressed**See Also:**[KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html), [KeyListener](http://docs.google.com/java/awt/event/KeyListener.html), [KeyboardFocusManager](http://docs.google.com/java/awt/KeyboardFocusManager.html), [DefaultKeyboardFocusManager](http://docs.google.com/java/awt/DefaultKeyboardFocusManager.html), [Component.processEvent(java.awt.AWTEvent)](http://docs.google.com/java/awt/Component.html#processEvent(java.awt.AWTEvent)), [Component.dispatchEvent(java.awt.AWTEvent)](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [Component.addKeyListener(java.awt.event.KeyListener)](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [Component.enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [Component.isShowing()](http://docs.google.com/java/awt/Component.html#isShowing())

### setKeySelectionManager

public void **setKeySelectionManager**([JComboBox.KeySelectionManager](http://docs.google.com/javax/swing/JComboBox.KeySelectionManager.html) aManager)

Sets the object that translates a keyboard character into a list selection. Typically, the first selection with a matching first character becomes the selected item.

### getKeySelectionManager

public [JComboBox.KeySelectionManager](http://docs.google.com/javax/swing/JComboBox.KeySelectionManager.html) **getKeySelectionManager**()

Returns the list's key-selection manager.

**Returns:**the KeySelectionManager currently in use

### getItemCount

public int **getItemCount**()

Returns the number of items in the list.

**Returns:**an integer equal to the number of items in the list

### getItemAt

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

Returns the list item at the specified index. If index is out of range (less than zero or greater than or equal to size) it will return null.

**Parameters:**index - an integer indicating the list position, where the first item starts at zero **Returns:**the Object at that list position; or null if out of range

### createDefaultKeySelectionManager

protected [JComboBox.KeySelectionManager](http://docs.google.com/javax/swing/JComboBox.KeySelectionManager.html) **createDefaultKeySelectionManager**()

Returns an instance of the default key-selection manager.

**Returns:**the KeySelectionManager currently used by the list**See Also:**[setKeySelectionManager(javax.swing.JComboBox.KeySelectionManager)](http://docs.google.com/javax/swing/JComboBox.html#setKeySelectionManager(javax.swing.JComboBox.KeySelectionManager))

### paramString

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

Returns a string representation of this JComboBox. 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/javax/swing/JComponent.html#paramString()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**a string representation of this JComboBox

### getAccessibleContext

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

Gets the AccessibleContext associated with this JComboBox. For combo boxes, the AccessibleContext takes the form of an AccessibleJComboBox. A new AccessibleJComboBox instance is created if necessary.

**Specified by:**[getAccessibleContext](http://docs.google.com/javax/accessibility/Accessible.html#getAccessibleContext()) in interface [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)**Overrides:**[getAccessibleContext](http://docs.google.com/javax/swing/JComponent.html#getAccessibleContext()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**an AccessibleJComboBox that serves as the AccessibleContext of this JComboBox

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JComboBox.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/javax/swing/JColorChooser.AccessibleJColorChooser.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JComboBox.AccessibleJComboBox.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JComboBox.html)    [**NO FRAMES**](http://docs.google.com/JComboBox.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#3as4poj) | [METHOD](#3o7alnk) |

[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).