
23

java.awt.peer Reference

23.1 ButtonPeer

Description

ButtonPeer is an interface that defines the basis for buttons.

Interface Definition

```
public abstract interface java.awt.peer.ButtonPeer
    extends java.awt.peer.ComponentPeer {

    // Interface Methods
    public abstract void setLabel (String label);
}
```

Interface Methods

setLabel

```
public abstract void setLabel (String label)
```

Parameters *label* New text for label of button's peer.

Description Changes the text of the label of button's peer.

See Also

ComponentPeer, String

23.2 CanvasPeer

Description

CanvasPeer is an interface that defines the basis for canvases.

Interface Definition

```
public abstract interface java.awt.peer.CanvasPeer
    extends java.awt.peer.ComponentPeer {
}
```

See Also

ComponentPeer

23.3 CheckboxMenuItemPeer



Description

CheckboxMenuItemPeer is an interface that defines the basis for checkbox menu items.

Interface Definition

```
public abstract interface java.awt.peer.CheckboxMenuItemPeer
    extends java.awt.peer.MenuItemPeer {

    // Interface Methods
    public abstract void setState (boolean condition);
}
```

Interface Methods

setState

```
public abstract void setState (boolean condition)
```

Parameters *condition* New state for checkbox menu item's peer.

Description Changes the state of checkbox menu item's peer.

See Also

MenuComponentPeer, MenuItemPeer

23.4 CheckboxPeer**Description**

CheckboxPeer is an interface that defines the basis for checkbox components.

Interface Definition

```

public abstract interface java.awt.peer.CheckboxPeer
    extends java.awt.peer.ComponentPeer {

    // Interface Methods
    public abstract void setCheckboxGroup (CheckboxGroup group);
    public abstract void setLabel (String label);
    public abstract void setState (boolean state);
}
  
```

Interface Methods**setCheckboxGroup**

```

public abstract void setCheckboxGroup (CheckboxGroup
group)
  
```

Parameters *group* New group to put the checkbox peer in.
 Description Changes the checkbox group to which the checkbox peer
 belongs; implicitly removes the peer from its old group, if any.

setLabel

```

public abstract void setLabel (String label)
  
```

Parameters *label* New text for label of checkbox's peer.
 Description Changes the text of the label of the checkbox's peer.

setState

```
public abstract void setState (boolean state)
```

Parameters *state* New state for the checkbox's peer.

Description Changes the state of the checkbox's peer.

See Also

CheckboxGroup, ComponentPeer, String

23.5 ChoicePeer

Description

ChoicePeer is an interface that defines the basis for choice components.

Interface Definition

```
public abstract interface java.awt.peer.ChoicePeer
    extends java.awt.peer.ComponentPeer {

    // Interface Methods
    public abstract void add (String item, int index); ★
    public abstract void addItem (String item, int position); ☆

    public abstract void remove (int index); ★
    public abstract void select (int position);
}
```

Interface Methods

add

```
public abstract void add (String item, int index) ★
```

Parameters *item* Text of the entry to add.

index Position in which to add the entry; position 0 is
the first entry in the list.

Description Adds a new entry to the available choices at the designated position.

addItem

```
public abstract void addItem (String item, int position)
☆
```

Parameters *item* Text of the entry to add.

position Position in which to add the entry; position 0 is
the first entry in the list.

Description Adds a new entry to the available choices at the designated position.

remove

```
public abstract void remove (int index) ★
```

Parameters *index* Position of the item to remove.

Description Removes an entry at the given position.

select

```
public abstract void select (int position)
```

Parameters *position* Position to make selected entry.

Description Makes the given entry the selected one for the choice's peer.

See Also

ComponentPeer, String

23.6 ComponentPeer

Description

ComponentPeer is an interface that defines the basis for all non-menu GUI peer interfaces.

Interface Definition

```
public abstract interface java.awt.peer.ComponentPeer {
    // Interface Methods
    public abstract int checkImage (Image image, int width, int height,
        ImageObserver observer);
    public abstract Image createImage (ImageProducer producer);
    public abstract Image createImage (int width, int height);
    public abstract void disable(); ☆
    public abstract void dispose();
    public abstract void enable(); ☆
    public abstract ColorModel getColorModel();
    public abstract FontMetrics getFontMetrics (Font f);
    public abstract Graphics getGraphics();
    public abstract Point getLocationOnScreen(); ★
    public abstract Dimension getMinimumSize(); ★
    public abstract Dimension getPreferredSize(); ★
    public abstract Toolkit getToolkit();
    public abstract boolean handleEvent (Event e);
    public abstract void hide(); ☆
    public abstract boolean isFocusTraversable(); ★
    public abstract Dimension minimumSize(); ☆
}
```

```
public abstract void paint (Graphics g);
public abstract Dimension preferredSize (); ☆
public abstract boolean prepareImage (Image image, int width, int height,
    ImageObserver observer);
public abstract void print (Graphics g);
public abstract void repaint (long tm, int x, int y, int width, int height);
public abstract void requestFocus();
public abstract void reshape (int x, int y, int width, int height); ☆
public abstract void setBackground (Color c);
public abstract void setBounds (int x, int y, int width, int height); ★
public abstract void setCursor (Cursor cursor); ★
public abstract void setEnabled (boolean b); ★
public abstract void setFont (Font f);
public abstract void setForeground (Color c);
public abstract void setVisible (boolean b); ★
public abstract void show(); ☆
}
```

Interface Methods

checkImage

```
public abstract int checkImage (Image image, int width,
int height, ImageObserver observer)
```

Parameters	<i>image</i>	Image to check.
	<i>width</i>	Horizontal size to which the image will be scaled.
	<i>height</i>	Vertical size to which the image will be scaled.
	<i>observer</i>	An ImageObserver to monitor image loading; normally, the object on which the image will be rendered.
Returns	ImageObserver flags ORed together indicating status.	
Description	Checks status of image construction.	

createImage

```
public abstract Image createImage (ImageProducer producer)
```

Parameters	<i>producer</i>	An object that implements the ImageProducer interface to create a new image.
Returns	Newly created image instance.	
Description	Creates an Image based upon an ImageProducer.	

```
public abstract Image createImage (int width, int height)
```

Parameters *width* Horizontal size for in-memory Image.

height Vertical size for in-memory Image.

Returns Newly created image instance.

Description Creates an in-memory Image for double buffering.

disable

```
public abstract void disable() ☆
```

Description Disables component so that it is unresponsive to user interactions. Replaced by `setEnabled(false)`.

dispose

```
public abstract void dispose()
```

Description Releases resources used by peer.

enable

```
public abstract void enable() ☆
```

Description Enables component so that it is responsive to user interactions. Replaced by `setEnabled(true)`.

getColorModel

```
public abstract ColorModel getColorModel()
```

Returns ColorModel used to display the current component.

getFontMetrics

```
public abstract FontMetrics getFontMetrics (Font f)
```

Parameters *f* A font whose metrics are desired.

Returns Font sizing information for the desired font.

getGraphics

```
public abstract Graphics getGraphics()
```

Throws *InternalException*

If acquiring a graphics context is unsupported

Returns Component's graphics context.

getLocationOnScreen

```
public abstract Point getLocationOnScreen() ★
```

Returns The location of the component in the screen's coordinate space.

getMinimumSize

```
public abstract Dimension getMinimumSize() ★
```

Returns The minimum dimensions of the component.

getPreferredSize

```
public abstract Dimension getPreferredSize() ★
```

Returns The preferred dimensions of the component.

getToolkit

```
public abstract Toolkit getToolkit()
```

Returns Toolkit of Component.

handleEvent

```
public abstract boolean handleEvent (Event e)
```

Parameters *e* Event instance identifying what caused the method to be called.

Returns true if the peer handled the event, false to propagate the event to the parent container.

Description High-level event handling routine.

hide

```
public abstract void hide() ☆
```

Description Hides the component. Replaced by `setVisible(false)`.

isFocusTraversable

```
public abstract boolean isFocusTraversable() ★
```

Returns true if the peer can be tabbed onto, false otherwise.

Description Determines if this peer is navigable using the keyboard.

minimumSize

```
public abstract Dimension minimumSize() ☆
```

Returns The minimum dimensions of the component. Replaced by `getMinimumSize()`.

paint

```
public abstract void paint (Graphics g)
```

Parameters *g* Graphics context of the component.

Description Draws something in graphics context.

preferredSize

```
public abstract Dimension preferredSize() ☆
```

Returns The preferred dimensions of the component. Replaced by `getPreferredSize()`.

prepareImage

```
public abstract boolean prepareImage (Image image, int
width,
int height, ImageObserver observer)
```

Parameters *image* Image to load.
 width Horizontal size to which the image will be
 scaled.
 height Vertical size to which the image will be scaled.
 observer An ImageObserver to monitor image loading;
 normally, the object on which the image will be
 rendered.

Returns true if the image has already loaded, false otherwise.

Description Forces the image to start loading.

print

```
public abstract void print (Graphics g)
```

Parameters *g* Graphics context of component.

Description Print something from the graphics context.

repaint

```
public abstract void repaint (long tm, int x, int y, int width, int height)
```

Parameters	<i>tm</i>	Millisecond delay allowed before repaint.
	<i>x</i>	Horizontal origin of bounding box to redraw.
	<i>y</i>	Vertical origin of bounding box to redraw.
	<i>width</i>	Width of bounding box to redraw.
	<i>height</i>	Height of bounding box to redraw.

Description Requests scheduler to redraw portion of component within a time period.

requestFocus

```
public abstract void requestFocus()
```

Description Requests this Component gets the input focus.

reshape

```
public abstract void reshape (int x, int y, int width, int height) ☆
```

Parameters	<i>x</i>	New horizontal position for component.
	<i>y</i>	New vertical position for component.
	<i>width</i>	New width for component.
	<i>height</i>	New height for component.

Description Relocates and resizes the component's peer. Replaced by `setBounds(int, int, int, int)`.

setBackground

```
public abstract void setBackground (Color c)
```

Parameters	<i>c</i>	New color for the background.
------------	----------	-------------------------------

Description Changes the background color of the component.

setBounds

```
public abstract void setBounds (int x, int y, int width, int height) ★
```

Parameters	<i>x</i>	New horizontal position for component.
	<i>y</i>	New vertical position for component.
	<i>width</i>	New width for component.
	<i>height</i>	New height for component.

Description Relocates and resizes the component's peer.

setCursor

```
public abstract void setCursor (Cursor cursor) ★
```

Parameters *cursor* New cursor.

Description Changes the cursor of the component.

setEnabled

```
public abstract void setEnabled (boolean b) ★
```

Parameters *b* true to enable the peer; false to disable it.

Description Enables or disables the peer.

setFont

```
public abstract void setFont (Font f)
```

Parameters *f* New font for the component.

Description Changes the font used to display text in the component.

setForeground

```
public abstract void setForeground (Color c)
```

Parameters *c* New foreground color for the component.

Description Changes the foreground color of the component.

setVisible

```
public abstract void setVisible (boolean b) ★
```

Parameters *b* true to show the peer; false to hide it.

Description Shows or hides the peer.

show

```
public abstract void show() ☆
```

Description Makes the peer visible. Replaced by `setVisible(true)`.

See Also

ButtonPeer, CanvasPeer, CheckboxPeer, ChoicePeer, Color, ColorModel, ContainerPeer, Cursor, Dimension, Event, Font, FontMetrics, Graphics, Image, ImageObserver, ImageProducer, LabelPeer, ListPeer, ScrollbarPeer, TextComponentPeer, Toolkit

23.7 *ContainerPeer*

Description

ContainerPeer is an interface that defines the basis for containers.

Interface Definition

```
public abstract interface java.awt.peer.ContainerPeer
    extends java.awt.peer.ComponentPeer {

    // Interface Methods
    public abstract void beginValidate(); ★
    public abstract void endValidate(); ★
    public abstract Insets getInsets(); ★
    public abstract Insets insets(); ☆
}
```

Interface Methods

beginValidate

```
public abstract void beginValidate() ★
```

Description Notifies the peer that the Container is going to validate its contents.

endValidate

```
public abstract void endValidate() ★
```

Description Notifies the peer that the Container is finished validating its contents.

getInsets

```
public Insets getInsets() ★
```

Returns Current Insets of container's peer.

insets

```
public Insets insets() ☆
```

Returns Current Insets of container's peer. Replaced by
getInsets().

See Also

ComponentPeer, Insets, PanelPeer, ScrollPanePeer, WindowPeer

23.8 *DialogPeer*

Description

DialogPeer is an interface that defines the basis for a dialog box.

Interface Definition

```
public abstract interface java.awt.peer.DialogPeer
    extends java.awt.peer.WindowPeer {

    // Interface Methods
    public abstract void setResizable (boolean resizable);
    public abstract void setTitle (String title);
}
```

Interface Methods

setResizable

```
public abstract void setResizable (boolean resizable)
```

Parameters *resizable* true if the dialog's peer should allow resizing;
false to prevent resizing.

Description Changes the resize state of the dialog's peer.

setTitle

```
public abstract void setTitle (String title)
```

Parameters *title* New title for the dialog's peer.

Description Changes the title of the dialog's peer.

See Also

FileDialogPeer, String, WindowPeer

23.9 *FileDialogPeer*

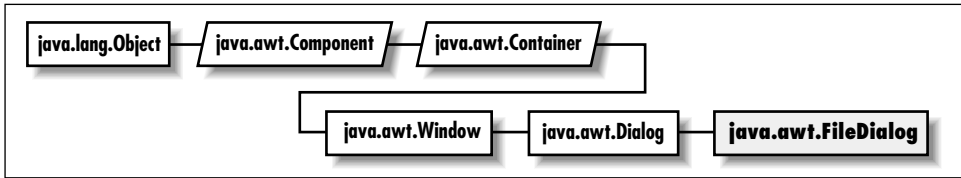
Description

FileDialogPeer is an interface that defines the basis for a file dialog box.

Interface Definition

```
public abstract interface java.awt.peer.FileDialogPeer
    extends java.awt.peer.DialogPeer {

    // Interface Methods
    public abstract void setDirectory (String directory);
    public abstract void setFile (String file);
    public abstract void setFilenameFilter (FilenameFilter filter);
}
```



```
}
```

Interface Methods

setDirectory

```
public abstract void setDirectory (String directory)
```

Parameters *directory* Initial directory for file dialog's peer.

Description Changes the directory displayed in the file dialog's peer.

setFile

```
public abstract void setFile (String file)
```

Parameters *file* Initial filename for the file dialog's peer.

Description Changes the default file selection for the file dialog's peer.

setFilenameFilter

```
public abstract void setFilenameFilter (FilenameFilter
filter)
```

Parameters *filter* Initial filter for file dialog's peer.

Description Changes the current filename filter of the file dialog's peer.

See Also

DialogPeer, FilenameFilter, String

23.10 *FontPeer* ★

Description

FontPeer is an interface that defines the basis for fonts.

Interface Definition

```
public abstract interface java.awt.peer.FontPeer {
}
```

See Also

ComponentPeer

23.11 FramePeer

Description

FramePeer is an interface that defines the basis for a frame.

Interface Definition

```
public abstract interface java.awt.peer.FramePeer
    extends java.awt.peer.WindowPeer {

    // Interface Methods
    public abstract void setIconImage (Image image);
    public abstract void setMenuBar (MenuBar bar);
    public abstract void setResizable (boolean resizable);
    public abstract void setTitle (String title);
}
```

Interface Methods

setIconImage

```
public abstract void setIconImage (Image image)
```

Parameters *image* New image to use for frame peer's icon.

Description Changes the icon associated with the frame's peer.

setMenuBar

```
public abstract void setMenuBar (MenuBar bar)
```

Parameters *bar* New MenuBar to use for the frame's peer.

Description Changes the menu bar of the frame.

setResizable

```
public abstract void setResizable (boolean resizable)
```

Parameters *resizable* true if the frame's peer should allow resizing,
false to prevent resizing.

Description Changes the resize state of the frame's peer.

setTitle

```
public abstract void setTitle (String title)
```

Parameters *title* New title to use for the frame's peer.

Description Changes the title of the frame's peer.

See Also

Image, MenuBar, String, WindowPeer

23.12 LabelPeer

Description

LabelPeer is an interface that defines the basis for label components.

Interface Definition

```
public abstract interface java.awt.peer.LabelPeer
    extends java.awt.peer.ComponentPeer {

    // Interface Methods
    public abstract void setAlignment (int alignment);
    public abstract void setText (String label);
}
```

Interface Methods**setAlignment**

```
public abstract void setAlignment (int alignment)
```

Parameters *alignment* New alignment for label's peer.

Description Changes the current alignment of label's peer.

setText

```
public abstract void setText (String label)
```

Parameters *label* New text for label's peer.

Description Changes the current text of label's peer.

See Also

ComponentPeer, String

23.13 *LightweightPeer* ★

Description

`LightweightPeer` is an interface that defines the basis for components that don't have a visual representation. When you directly subclass `Component` or `Container`, a `LightweightPeer` is used.

Interface Definition

```
public abstract interface java.awt.peer.LightweightPeer
    extends java.awt.peer.ComponentPeer {
}
```

See Also

`ComponentPeer`

23.14 *ListPeer*

Description

`ListPeer` is an interface that defines the basis for list components.

Interface Definition

```
public abstract interface java.awt.peer.ListPeer
    extends java.awt.peer.ComponentPeer {

    // Interface Methods
    public abstract void add (String item, int index); ★
    public abstract void addItem (String item, int index); ☆
    public abstract void clear(); ☆
    public abstract void delItems (int start, int end);
    public abstract void deselect (int index);
    public abstract Dimension getMinimumSize (int rows); ★
    public abstract Dimension getPreferredSize (int rows); ★
    public abstract int[] getSelectedIndexes();
    public abstract void makeVisible (int index);
    public abstract Dimension minimumSize (int rows); ☆
    public abstract Dimension preferredSize (int rows); ☆
    public abstract void removeAll(); ★
    public abstract void select (int position);
    public abstract void setMultipleMode (boolean b); ★
    public abstract void setMultipleSelections (boolean value); ☆
}
```

Interface Methods

add

```
public abstract void add (String item, int index) ★
```

Parameters	<i>item</i>	Text of an entry to add to the list.
	<i>index</i>	Position in which to add the entry; position 0 is the first entry in the list.
Description	Adds a new entry to the available choices of the list's peer at the designated position.	

addItem

```
public abstract void addItem (String item, int index) ☆
```

Parameters	<i>item</i>	Text of an entry to add to the list.
	<i>index</i>	Position in which to add the entry; position 0 is the first entry in the list.
Description	Adds a new entry to the available choices of the list's peer at the designated position. Replaced by <code>add(String, int)</code> .	

clear

```
public abstract void clear() ☆
```

Description	Clears all the entries out of the list's peer. Replaced by <code>removeAll()</code> .	
-------------	---	--

delItems

```
public abstract void delItems (int start, int end)
```

Parameters	<i>start</i>	Starting position of entries to delete.
	<i>end</i>	Ending position of entries to delete.
Description	Removes a set of entries from the list's peer.	

deselect

```
public abstract void deselect (int index)
```

Parameters	<i>index</i>	Position to deselect.
Description	Deselects entry at designated position, if selected.	

getMinimumSize

```
public abstract Dimension getMinimumSize (int rows) ★
```

Parameters *rows* Number of rows within list's peer to size.
Returns The minimum dimensions of a list's peer of the given size.

getPreferredSize

```
public abstract Dimension getPreferredSize (int rows) ★
```

Parameters *rows* Number of rows within list's peer to size.
Returns The preferred dimensions of a list's peer of the given size.

getSelectedIndexes

```
public abstract int[] getSelectedIndexes ()
```

Returns Array of positions of currently selected entries in list's peer.

makeVisible

```
public abstract void makeVisible (int index)
```

Parameters *index* Position to make visible on screen.
Description Ensures an item is displayed on the screen in the list's peer.

minimumSize

```
public abstract Dimension minimumSize (int rows) ☆
```

Parameters *rows* Number of rows within list's peer to size.
Returns The minimum dimensions of a list's peer of the given size.
 Replaced by `getMinimumSize(int)`.

preferredSize

```
public abstract Dimension preferredSize (int rows) ☆
```

Parameters *rows* Number of rows within list's peer to size.
Returns The preferred dimensions of a list's peer of the given size.
 Replaced by `getPreferredSize(int)`.

removeAll

```
public abstract void removeAll () ★
```

Description Clears all the entries out of the list's peer.

select

```
public abstract void select (int position)
```

Parameters *position* Position to select; 0 indicates the first item in the list.

Description Makes the given entry the selected item for the list's peer; deselects other selected entries if multiple selections are not enabled.

setMultipleMode

```
public abstract void setMultipleMode (boolean value) ★
```

Parameters *value* true to allow multiple selections within the list's peer; false to disallow multiple selections.

Description Changes list peer's selection mode.

setMultipleSelections

```
public abstract void setMultipleSelections (boolean value)
```

☆

Parameters *value* true to allow multiple selections within the list's peer; false to disallow multiple selections.

Description Changes list peer's selection mode. Replaced by `setMultipleMode(boolean)`.

See Also

ComponentPeer, Dimension, String

23.15 MenuBarPeer

**Description**

MenuBarPeer is an interface that defines the basis for menu bars.

Interface Definition

```
public abstract interface java.awt.peer.MenuBarPeer
    extends java.awt.peer.MenuComponentPeer {
```

```
    // Interface Methods
```

```
    public abstract void addHelpMenu (Menu m);
```

```

public abstract void addMenu (Menu m);
public abstract void delMenu (int index);
}

```

Interface Methods

addHelpMenu

```
public abstract void addHelpMenu (Menu m)
```

Parameters *m* Menu to designate as the help menu with the menu bar's peer.

Description Sets a particular menu to be the help menu of the menu bar's peer.

addMenu

```
public abstract void addMenu (Menu m)
```

Parameters *m* Menu to add to the menu bar's peer

Description Adds a menu to the menu bar's peer.

delMenu

```
public abstract void delMenu (int index)
```

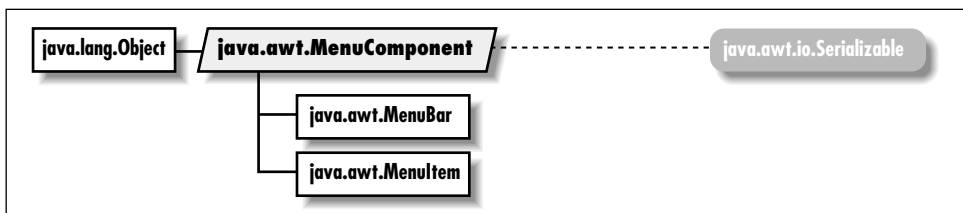
Parameters *index* Menu position to delete from the menu bar's peer.

Description Deletes a menu from the menu bar's peer.

See Also

Menu, MenuComponentPeer

23.16 MenuComponentPeer



Description

MenuComponentPeer is an interface that defines the basis for all menu GUI peer interfaces.

Interface Definition

```
public abstract interface java.awt.peer.MenuComponentPeer {  
  
    // Interface Methods  
    public abstract void dispose();  
}
```

Interface Methods

dispose

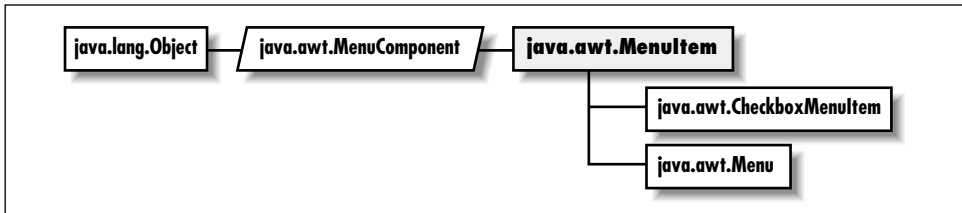
```
public abstract void dispose()
```

Description Releases resources used by peer.

See Also

MenuBarPeer, MenuItemPeer

23.17 MenuItemPeer



Description

MenuBarPeer is an interface that defines the basis for menu bars.

Interface Definition

```
public abstract interface java.awt.peer.MenuItemPeer  
    extends java.awt.peer.MenuComponentPeer {  
  
    // Interface Methods  
    public abstract void disable(); ☆  
    public abstract void enable(); ☆  
    public abstract void setEnabled (boolean b); ★  
    public abstract void setLabel (String label);  
}
```

Interface Methods

disable

```
public abstract void disable() ☆
```

Description Disables the menu item's peer so that it is unresponsive to user interactions. Replaced by `setEnabled(false)`.

enable

```
public abstract void enable() ☆
```

Description Enables the menu item's peer so that it is responsive to user interactions. Replaced by `setEnabled(true)`.

setEnabled

```
public abstract void setEnabled (boolean b) ★
```

Parameters *b* true to enable the peer; false to disable it.

Description Enables or disables the menu item's peer.

setLabel

```
public abstract void setLabel (String label)
```

Parameters *label* New text to appear on the menu item's peer.

Description Changes the label of the menu item's peer.

See Also

`CheckboxMenuItemPeer`, `MenuComponentPeer`, `MenuPeer`, `String`

23.18 MenuPeer

Description

`MenuPeer` is an interface that defines the basis for menus.

Interface Definition

```
public abstract interface java.awt.peer.MenuPeer
    extends java.awt.peer.MenuItemPeer {

    // Interface Methods
    public abstract void addItem (MenuItem item);
    public abstract void addSeparator();
    public abstract void delItem (int index);
}
```

Interface Methods

addItem

```
public abstract void addItem (MenuItem item)
```

Parameters *item* MenuItem to add to the menu's peer

Description Adds a menu item to the menu's peer.

addSeparator

```
public abstract void addSeparator()
```

Description Adds a menu separator to the menu's peer.

delItem

```
public abstract void delItem (int index)
```

Parameters *index* MenuItem position to delete from the menu's peer.

Description Deletes a menu item from the menu's peer.

See Also

MenuItem, MenuItemPeer

23.19 PanelPeer

Description

PanelPeer is an interface that defines the basis for a panel.

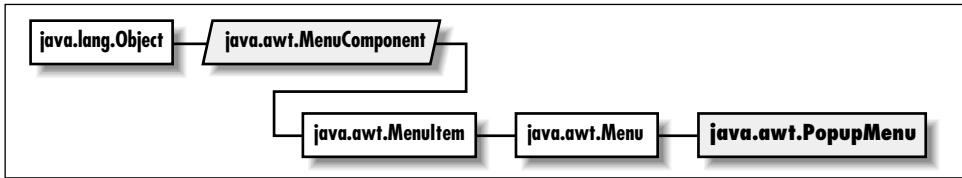
Interface Definition

```
public abstract interface java.awt.peer.PanelPeer
    extends java.awt.peer.ContainerPeer {
}
```

See Also

ContainerPeer

23.20 PopupMenuPeer ★



Description

PopupMenuPeer is an interface that defines the basis for a popup menu.

Interface Definition

```

public abstract interface java.awt.peer.PopupMenuPeer
    extends java.awt.peer.MenuPeer {

    // Interface Methods
    public abstract void show (Event e);
}
  
```

Interface Methods

show

```
public abstract void show (Event e)
```

Parameters *e* A mouse down event that begins the display of the popup menu.

Description Shows the peer at the location encapsulated in *e*.

See Also

Event, MenuPeer

23.21 ScrollbarPeer



Description

ScrollbarPeer is an interface that defines the basis for scrollbar components.

Interface Definition

```
public abstract interface java.awt.peer.ScrollbarPeer
    extends java.awt.peer.ComponentPeer {

    // Interface Methods
    public abstract void setLineIncrement (int amount);
    public abstract void setPageIncrement (int amount);
    public abstract void setValues (int value, int visible, int minimum, int maximum);
}
```

Interface Methods

setLineIncrement

```
public abstract void setLineIncrement (int amount)
```

Parameters *amount* New line increment amount.

Description Changes the line increment amount for the scrollbar's peer.

setPageIncrement

```
public abstract void setPageIncrement (int amount)
```

Parameters *amount* New paging increment amount.

Description Changes the paging increment amount for the scrollbar's peer.

setValues

```
public abstract void setValues (int value, int visible,
int minimum, int maximum)
```

Parameters *value* New value for the scrollbar's peer.

visible New slider width.

minimum New minimum value for the scrollbar's peer.

maximum New maximum value for the scrollbar's peer.

Description Changes the settings of the scrollbar's peer to the given amounts.

See Also

ComponentPeer

23.22 ScrollPanePeer ★



Description

ScrollPanePeer is an interface that defines the basis for a scrolling container.

Interface Definition

```

public abstract interface java.awt.peer.ScrollPanePeer
    extends java.awt.peer.ContainerPeer {

    // Interface Methods
    public abstract void childResized (int w, int h);
    public abstract int getHScrollbarHeight();
    public abstract int getVScrollbarWidth();
    public abstract void setScrollPosition (int x, int y);
    public abstract void setUnitIncrement (Adjustable adj, int u);
    public abstract void setValue (Adjustable adj, int v);
}
  
```

Interface Methods

childResized

```
public abstract void childResized (int w, int h)
```

Parameters *w* The new child width.
 h The new child height.

Description Tells the peer that the child has a new size.

getHScrollbarHeight

```
public abstract int getHScrollbarHeight()
```

Returns Height that a horizontal scrollbar would occupy.

Description The height is returned regardless of whether the scrollbar is showing or not.

getVScrollbarWidth

```
public abstract int getVScrollbarWidth()
```

Returns Width that a vertical scrollbar would occupy.

Description The width is returned regardless of whether the scrollbar is showing or not.

setScrollPosition

```
public abstract void setScrollPosition (int x, int y)
```

Parameters *x* The new horizontal position.
y The new vertical position.

Description Changes the coordinate of the child component that is displayed at the origin of the ScrollPanePeer.

setUnitIncrement

```
public abstract void setUnitIncrement (Adjustable adj, int u)
```

Parameters *adj* The Adjustable object to change.
u The new value.

Description Changes the unit increment of the given Adjustable object.

setValue

```
public abstract void setValue (Adjustable adj, int v)
```

Parameters *adj* The Adjustable object to change.
v The new value.

Description Changes the value of the given Adjustable object.

See Also

Adjustable, ContainerPeer, Scrollbar

23.23 TextAreaPeer



Description

TextAreaPeer is an interface that defines the basis for text areas.

Interface Definition

```
public abstract interface java.awt.peer.TextAreaPeer
    extends java.awt.peer.TextComponentPeer {

    // Interface Methods
    public abstract Dimension getMinimumSize (int rows, int columns); ★
    public abstract Dimension getPreferredSize (int rows, int columns); ★
    public abstract void insert (String string, int position); ★
    public abstract void insertText (String string, int position); ☆
    public abstract Dimension minimumSize (int rows, int columns); ☆
    public abstract Dimension preferredSize (int rows, int columns); ☆
    public abstract void replaceRange (String string, int startPosition, int endPosition); ☆
    public abstract void replaceText (String string, int startPosition, int endPosition); ☆
}
```

Interface Methods

getMinimumSize

```
public abstract Dimension getMinimumSize
(int rows, int columns) ★
```

Parameters *rows* Number of rows within the text area's peer.
 columns Number of columns within the text area's peer.

Returns The minimum dimensions of a text area's peer of the given size.

getPreferredSize

```
public abstract Dimension getPreferredSize
(int rows, int columns) ★
```

Parameters *rows* Number of rows within the text area's peer.
 columns Number of columns within the text area's peer.

Returns The preferred dimensions of a text area's peer of the given size.

insert

```
public abstract void insert ★ (String string, int position)
★
```

Parameters *string* Content to place within the text area's peer.
 position Location at which to insert the content.

Description Places additional text within the text area's peer.

insertText

```
public abstract void insertText (String string, int  
position) ☆
```

Parameters *string* Content to place within the text area's peer.
 position Location at which to insert the content.

Description Places additional text within the text area's peer. Replaced by
 insert(String, int).

minimumSize

```
public abstract Dimension minimumSize (int rows, int  
columns) ☆
```

Parameters *rows* Number of rows within the text area's peer.
 columns Number of columns within the text area's peer.

Returns The minimum dimensions of a text area's peer of the given
 size. Replaced by getMinimumSize(int, int).

preferredSize

```
public abstract Dimension preferredSize (int rows, int  
columns) ☆
```

Parameters *rows* Number of rows within the text area's peer.
 columns Number of columns within the text area's peer.

Returns The preferred dimensions of a text area's peer of the given size.
 Replaced by getPreferredSize(int, int).

replaceRange

```
public abstract void replaceRange (String string,  
int startPosition, int endPosition) ★
```

Parameters *string* New content to place in the text area's peer.
 startPosition Starting position of the content to replace.
 endPosition Ending position of the content to replace.

Description Replaces a portion of the text area peer's content with the
 given text.

replaceText

```
public abstract void replaceText (String string,  
int startPosition, int endPosition) ☆
```

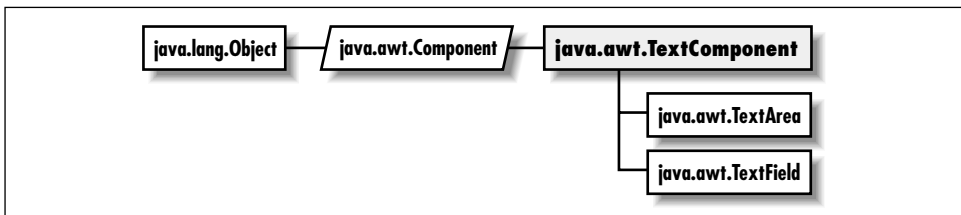
Parameters *string* New content to place in the text area's peer.

- startPosition* Starting position of the content to replace.
- endPosition* Ending position of the content to replace.
- Description Replaces a portion of the text area peer's content with the given text. Replaced by `replaceRange(String, int, int)`.

See Also

Dimension, String, TextComponentPeer

23.24 TextComponentPeer



Description

TextComponentPeer is an interface that defines the basis for text components.

Interface Definition

```

public abstract interface java.awt.peer.TextComponentPeer
    extends java.awt.peer.ComponentPeer {

    // Interface Methods
    public abstract int getCaretPosition(); ★
    public abstract int getSelectionEnd();
    public abstract int getSelectionStart();
    public abstract String getText();
    public abstract void select (int selectionStart, int selectionEnd);
    public abstract void setCaretPosition (int pos); ★
    public abstract void setEditable (boolean state);
    public abstract void setText (String text);
}
  
```

Interface Methods

getCaretPosition

```
public abstract int getCaretPosition() ★
```

Returns The current position of the caret (text cursor).

getSelectionEnd

```
public abstract int getSelectionEnd()
```

Returns The ending cursor position of any selected text.

getSelectionStart

```
public abstract int getSelectionStart()
```

Returns The initial position of any selected text.

getText

```
public abstract String getText()
```

Returns The current contents of the text component's peer.

select

```
public abstract void select (int selectionStart, int  
selectionEnd)
```

Parameters *selectionStart* Beginning position of the text to select.
 selectionEnd Ending position of the text to select.

Description Selects text in the text component's peer.

selectCaretPosition

```
public abstract void selectCaretPosition (int pos)
```

Parameters *pos* New caret position.

Description Changes the position of the caret (text cursor).

setEditable

```
public abstract void setEditable (boolean state)
```

Parameters *state* true if the user can change the contents of the
 text component's peer (i.e., true to make the
 peer editable); false to make the peer read-
 only.

Description Allows you to change the current editable state of the text com-
 ponent's peer.

setText

```
public abstract void setText (String text)
```

Parameters *text* New text for the text component's peer .

Description Sets the content of the text component's peer.

See Also

ComponentPeer, String, TextAreaPeer, TextFieldPeer

23.25 *TextFieldPeer*

**Description**

TextFieldPeer is an interface that defines the basis for text fields.

Interface Definition

```
public abstract interface java.awt.peer.TextFieldPeer
    extends java.awt.peer.TextComponentPeer {

    // Interface Methods
    public abstract Dimension getMinimumSize (int rows, int columns); ★
    public abstract Dimension getPreferredSize (int rows, int columns); ★
    public abstract Dimension minimumSize (int rows, int columns); ☆
    public abstract Dimension preferredSize (int rows, int columns); ☆
    public abstract void setEchoChar (char echoChar); ★
    public abstract void setEchoCharacter (char c); ☆
}
```

Interface Methods**getMinimumSize**

```
public abstract Dimension getMinimumSize (int rows) ★
```

Parameters *rows* Number of rows within the text field's peer.

Returns The minimum dimensions of a text field's peer of the given size.

getPreferredSize

public abstract Dimension getPreferredSize (int rows) ★

Parameters *rows* Number of rows within the text field's peer.
Returns The preferred dimensions of a text field's peer of the given size.

minimumSize

public abstract Dimension minimumSize (int rows) ☆

Parameters *rows* Number of rows within the text field's peer.
Returns Replaced by `getMinimumSize(int)`.

preferredSize

public abstract Dimension preferredSize (int rows) ☆

Parameters *rows* Number of rows within the text field's peer.
Returns Replaced by `getPreferredSize(int)`.

setEchoChar

public abstract void setEchoChar (char c) ★

Parameters *c* The character to display for all input.
Description Changes the character that is displayed to the user for every character he or she types in the text field.

setEchoCharacter

public abstract void setEchoCharacter (char c) ☆

Parameters *c* The character to display for all input.
Description Replaced by `setEchoChar(char)`.

See Also

Dimension, TextComponentPeer

23.26 WindowPeer

Description

WindowPeer is an interface that defines the basis for a window.

Interface Definition

```
public abstract interface java.awt.peer.WindowPeer
    extends java.awt.peer.ContainerPeer {

    // Interface Methods
    public abstract void toBack();
    public abstract void toFront();
}
```

Interface Methods

toBack

```
public abstract void toBack()
```

Description Puts the window's peer in the background of the display.

toFront

```
public abstract void toFront()
```

Description Brings the window's peer to the foreground of the display.

See Also

ContainerPeer, DialogPeer, FramePeer