NSColorPickingDefault Protocol Reference

Cocoa > User Experience

2007-04-27

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

NSColorPickingDefault Protocol Reference

Adopted by	NSColorPicker
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Color Programming Topics for Cocoa
Declared in	NSColorPicking.h

Overview

The `NSColorPickingDefault` protocol, together with the `NSColorPickingCustom` protocol, provides an interface for adding color pickers—custom user interfaces for color selection—to an application's `NSColorPanel` instance. The `NSColorPickingDefault` protocol provides basic behavior for a color picker. The `NSColorPickingCustom` protocol provides implementation-specific behavior.

Tasks

Creating Color Pickers

`– initWithPickerMask:colorPanel:` (page 8)
Initializes the receiver with a given color panel and its mode.

Configuring Color Pickers

`– setMode:` (page 10)
Specifies the receiver's mode.

`– insertNewButtonImage:in:` (page 9)
Sets the image of a given button cell.

`– provideNewButtonImage` (page 9)
Provides the image of the button used to select the receiver in the color panel.

`– minContentSize` (page 9)
Indicates the receiver's minimum content size.

- `buttonToolTip` (page 7)
  Provides the toolbar button help tag.

Handling Events

- `alphaControlAddedOrRemoved:` (page 6)
  Sent when the color panel's opacity controls have been hidden or displayed.
- `viewSizeChanged:` (page 10)
  Tells the recever when the color panel's view size changes in a way that might affect the color picker.

Managing Color Lists

- `attachColorList:` (page 6)
  Tells the receiver to attach the given color list, if it isn't already displaying the list.
- `detachColorList:` (page 7)
  Tells the receiver to detach the given color list, unless the receiver isn't displaying the list.

Instance Methods

alphaControlAddedOrRemoved:

Sent when the color panel's opacity controls have been hidden or displayed.

```
- (void)alphaControlAddedOrRemoved:(id)sender
```

Parameters

sender

The color panel sending the message.

Discussion

This method is invoked automatically when the opacity slider of the `NSColorPanel` is added or removed; you never invoke this method directly.

If the color picker has its own opacity controls, it should hide or display them, depending on whether the sender's `showsAlpha` method returns `NO` or `YES`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSColorPicking.h`

attachColorList:

Tells the receiver to attach the given color list, if it isn't already displaying the list.

```
- (void)attachColorList:(NSColorList *)colorList
```

Parameters

`colorList`

The color list to display.

Discussion

You never invoke this method; it's invoked automatically by the `NSColorPanel` object when its `attachColorList:` method is invoked. Because the `NSColorPanel` list mode manages `NSColorList` objects, this method need only be implemented by a custom color picker that manages `NSColorList` objects itself.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- detachColorList:` (page 7)

Declared In

`NSColorPicking.h`

buttonToolTip

Provides the toolbar button help tag.

```
- (NSString *)buttonToolTip
```

Return Value

Help tag text.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSColorPicking.h`

detachColorList:

Tells the receiver to detach the given color list, unless the receiver isn't displaying the list.

```
- (void)detachColorList:(NSColorList *)colorList
```

Parameters

`colorList`

The color list to detach.

Discussion

You never invoke this method; it's invoked automatically by the `NSColorPanel` object when its `detachColorList:` method is invoked. Because the `NSColorPanel` list mode manages `NSColorList` objects, this method need only be implemented by a custom color picker that manages `NSColorList` objects itself.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- attachColorList:` (page 6)

Declared In

`NSColorPicking.h`

initWithPickerMask:colorPanel:

Initializes the receiver with a given color panel and its mode.

```
- (id)initWithPickerMask:(NSUInteger)panelModes colorPanel:(NSColorPanel
    *)owningColorPanel
```

Parameters

panelModes

A mask indicating the various color picker modes supported by the color panel. This is determined by the argument to the `NSColorPanel` method `setPickerMask:`. If it has not been set, *panelModes* is `NSColorPanelAllModesMask`. If your color picker supports any additional modes, you should invoke the `setPickerMask:` method when your application initializes to notify the `NSColorPanel` class. The standard mode constants are defined in "Choosing the Color Pickers in a Color Panel".

owningColorPanel

The color panel than owns the receiver.

Return Value

If your color picker responds to any of the modes represented in *panelModes*, it should perform its initialization and return an initialized color picker. Color pickers that do so have their buttons inserted in the color panel and continue to receive messages from the panel as the user manipulates it. If the color picker doesn't respond to any of the modes represented in *panelModes*, it should do nothing and return `nil`.

Discussion

This method is sent by the `NSColorPanel` to all implementors of the color-picking protocols when the application's color panel is first initialized. In order for your color picker to receive this message, it must have a bundle in your application's "ColorPickers" directory (described in "Color Picker Bundles").

This method should examine the mask and determine whether it supports any of the modes included there. You may also check the value in *mask* to enable or disable any subpickers or optional controls implemented by your color picker. Your color picker may also retain *owningColorPanel* in an instance variable for future communication with the color panel.

This method is provided to initialize your color picker; however, much of a color picker's initialization may be done lazily through the `NSColorPickingCustom` protocol's `provideNewView:` method.

Availability

Available in Mac OS X v10.0 and later.

See Also

`+ setPickerMask:` (NSColorPanel class)

Declared In

`NSColorPicking.h`

insertNewButtonImage:in:

Sets the image of a given button cell.

```
- (void)insertNewButtonImage:(NSImage *)newButtonImage in:(NSButtonCell *)buttonCell
```

Parameters

newButtonImage

The image to set for the button cell.

buttonCell

The NSButtonCell object that lets the user choose the picker from the color panel—the color picker's representation in the NSMatrix of the NSColorPanel.

Discussion

This method should perform application-specific manipulation of the image before it's inserted and displayed by the button cell.

Availability

Available in Mac OS X v10.0 and later.

See Also

- provideNewButtonImage (page 9)

Declared In

NSColorPicking.h

minContentSize

Indicates the receiver's minimum content size.

```
- (NSSize)minContentSize
```

Discussion

The receiver does not allow a size smaller than minContentSize.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSColorPicking.h

provideNewButtonImage

Provides the image of the button used to select the receiver in the color panel.

```
- (NSImage *)provideNewButtonImage
```

Return Value

The image for the mode button the user uses to select this picker in the color panel; that is, the color picker's representation in the NSMatrix of the NSColorPanel.

This image is the same one the color panel uses as an argument when sending the insertNewButtonImage:in: (page 9) message.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSColorPicking.h`

setMode:

Specifies the receiver's mode.

```
- (void)setMode:(NSColorPanelMode)mode
```

Parameters

mode

The color picker mode. The available modes are described in "Choosing the Color Pickers in a Color Panel".

Discussion
This method is invoked by the `NSColorPanel` method `setMode:` method to ensure the color picker reflects the current mode. For example, invoke this method during color picker initialization to ensure that all color pickers are restored to the mode the user left them in the last time an `NSColorPanel` was used.

Most color pickers have only one mode and thus don't need to do any work in this method. An example of a color picker that uses this method is the slider picker, which can choose from one of several submodes depending on the value of *mode*.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSColorPicking.h`

viewSizeChanged:

Tells the recever when the color panel's view size changes in a way that might affect the color picker.

```
- (void)viewSizeChanged:(id)sender
```

Parameters

sender

The `NSColorPanel` that contains the color picker.

Discussion
Use this method to perform special preparation when resizing the color picker's view. Because this method is invoked only as appropriate, it's better to implement this method than to override the method `superviewSizeChanged:` for the `NSView` in which the color picker's user interface is contained.

Availability
Available in Mac OS X v10.0 and later.

See Also
`- provideNewView:` (`NSColorPickingCustom` protocol)

Declared In

`NSColorPicking.h`

Document Revision History

This table describes the changes to *NSColorPickingDefault Protocol Reference*.

Date	Notes
2007-04-27	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Index