NSColorPickingCustom

Adopted By:　　　　NSColorPicker

Declared In:　　　　AppKit/NSColorPicking.h

Protocol Description

Together with the NSColorPickingDefault protocol, NSColorPickingCustom provides a way to add color pickers—custom user interfaces for color selection—to an application's NSColorPanel. The NSColorPickingDefault protocol provides basic behavior for a color picker. The NSColorPicker class adopts the NSColorPickingDefault protocol. The easiest way to implement a color picker is to create a subclass of NSColorPicker and implement the NSColorPickingCustom protocol for this new class.

Note: All of NSColorPickingCustom's methods must be implemented by the custom color picker.

Here are the standard color picking modes and mode constants (defined in **AppKit/NSColorPanel.h**):

Mode	Color Mode Constant
Grayscale-Alpha	NSGrayModeColorPanel
Red-Green-Blue	NSRGBModeColorPanel
Cyan-Yellow-Magenta-Black	NSCMYKModeColorPanel
Hue-Saturation-Brightness	NSHSBModeColorPanel
Custom palette	NSCustomPaletteModeColorPanel
Custom color list	NSColorListModeColorPanel
Color wheel	NSWheelModeColorPanel

In grayscale-alpha, red-green-blue, cyan-magenta-yellow-black, and hue-saturation-brightness modes, the user adjusts colors by manipulating sliders. In the custom palette mode, the user can load an NSImage file (TIFF or EPS) into the NSColorPanel, then select colors from the image. In custom color list mode, the user can create and load lists of named colors. The two custom modes provide NSPopUpLists for loading and saving files. Finally, color wheel mode provides a simplified control for selecting colors.

If your color picker includes submodes, you should define a unique value for each submode. As an example, the slider picker has four values defined in the above list (NSGrayModeColorPanel, NSRGBModeColorPanel, NSCMYKModeColorPanel, and NSHSBModeColorPanel)—one for each of its submodes.

See also: NSColorPickingDefault, NSColorPicker (class)

Method Types

Setting the Current Color	– setColor:
Getting the Mode	– currentMode
	– supportsMode:
Getting the View	– provideNewView:

Instance Methods

currentMode

– (int)**currentMode**

Returns the color picker's current mode (or submode, if applicable). The returned value should be unique to your color picker. See this protocol description's list of the unique values for the standard color pickers used by the Application Kit.

See also: – supportsMode:

provideNewView:

– (NSView *)**provideNewView:**(BOOL)*initialRequest*

Returns the view containing the color picker's user interface. This message is sent to the color picker whenever the color panel attempts to display it. This may be when the panel is first presented, when the user switches pickers, or when the picker is switched through API. The argument *initialRequest* is YES only when this method is first invoked for your color picker. If *initialRequest* is YES, the method should perform any initialization required (such as lazily loading a nib file, initializing the view, or performing any other custom initialization required for your picker). The NSView returned by this method should be set to automatically resize both its width and height.

setColor:

– (void)**setColor:**(NSColor *)*color*

Adjusts the color picker to make *color* the currently selected color. This method is invoked on the current color picker each time NSColorPanel's **setColor:** method is invoked. If *color* is actually different from the color picker's color (as it would be if, for example, the user dragged a color into NSColorPanel's color well), this method could be used to update the color picker's color to reflect the change.

supportsMode:

– (BOOL)**supportsMode:**(int)*mode*

Returns whether or not the receiver supports the specified picking mode. This method is invoked when the NSColorPanel's is first initialized: It is used to attempt to restore the user's previously selected mode. It is also invoked by NSColorPanel's **setMode:** method to find the color picker that supports a particular mode. See this protocol description's list of the unique mode values for the standard color pickers used by the Application Kit.

See also: – currentMode: