NSFontPanel

Inherits From:	NSPanel : NSWindow : NSResponder : NSObject
Conforms To:	NSCoding (NSResponder) NSObject (NSObject)
Declared In:	AppKit/NSFontPanel.h

Class Description

The NSFontPanel class implements the Font Panel—a user-interface object that displays a list of available fonts, letting the user preview them and change the font used to display text. The actual changes are made through conversion messages sent to the shared NSFontManager instance. There's only one Font Panel for each application.

In general, you add the facilities of the Font Panel to your application, along with the NSFontManager and the Font menu, through Interface Builder. You do this by dragging a Font menu into one of your application's menus. At run time, when the user chooses the Font Panel command for the first time, the Font Panel object is created and hooked into the font conversion system. You can also create (or access) the Font Panel using the **sharedFontPanel** class method.

You can add a custom view object to an NSFontPanel using **setAccessoryView:**, or limit the fonts display by assigning a delegate to the application's font manager object. If you want the NSFontManager to instantiate the Font Panel from some class other than NSFontPanel, use NSFontManager's **setFontPanelFactory:** class method. See the NSFontManager class specification for more information on using the font conversion system.

Method Types

Getting the Font Panel	+ sharedFontPanel + sharedFontPanelExists
Enabling font changes	– setEnabled: – isEnabled
Updating the Font Panel	– setPanelFont:isMultiple:
Converting fonts	– panelConvertFont:
Working in modal loops	– worksWhenModal:

Setting an accessory view	– setAccessoryView: – accessoryView

Class Methods

sharedFontPanel

+ (NSFontPanel *)**sharedFontPanel**

Returns the single NSFontPanel instance for the application, creating it if necessary.

See also: + **sharedFontPanelExists**, + **setFontPanelFactory:** (NSFontManager)

sharedFontPanelExists

+ (BOOL)**sharedFontPanelExists**

Returns YES if the shared Font Panel has been created, NO if it hasn't.

See also: + **sharedFontPanel**

Instance Methods

accessoryView

– (NSView *)**accessoryView**

Returns the receiver's accessory view.

See also: – **setAccessoryView:**

isEnabled

– (BOOL)**isEnabled**

Returns YES if the receiver's Set button is enabled, NO if it isn't. The receiver continues to reflect the font of the selection for cooperating text objects regardless of this setting.

See also: – **setEnabled:**

panelConvertFont:

– (NSFont *)**panelConvertFont:**(NSFont *)*aFont*

Converts *aFont* using the settings in the receiver, with the aid of the shared NSFontManager if necessary, and returns the converted font. If *aFont* can't be converted it's returned unchanged.

For example, if *aFont* is Helvetica Oblique 12.0 point and the user has selected the Times font family (and nothing else) in the Font Panel, the font returned is Times Italic 12.0 point.

See also: – **convertFont:** (NSFontManager)

setAccessoryView:

– (void)**setAccessoryView:**(NSView *)*aView*

Establishes *aView* as the receiver's accessory view, allowing you to add custom controls to your application's Font Panel without having to create a subclass.

See also: – **accessoryView**

setEnabled:

– (void)**setEnabled:**(BOOL)*flag*

Controls whether the receiver's Set button is enabled. If *flag* is YES the Set button is enabled; if *flag* is NO it's disabled. The receiver continues to reflect the font of the selection for cooperating text objects regardless of this setting.

See also: – **isEnabled**

setPanelFont:isMultiple:

– (void)**setPanelFont:**(NSFont *)*aFont* **isMultiple:**(BOOL)*flag*

Sets the selected font in the receiver to *aFont* if *flag* is NO, otherwise selects no font and displays a message in the preview area indicating that multiple fonts are selected. You normally don't use this method directly; instead, you send **setSelectedFont:isMultiple:** to the shared NSFontManager, which in turn invokes this method.

worksWhenModal

– (BOOL)**worksWhenModal**

Returns YES, regardless of the setting established using the NSPanel method **setWorksWhenModal:**. This allows fonts to be changed in modal windows and panels.

See also: – **worksWhenModal** (NSWindow, NSPanel)