NSPrintPanel

Inherits From:	NSPanel : NSObject
Conforms To:	NSObject (NSObject)
Declared In:	AppKit/NSPrintPanel.h

Note: On Mach platforms, NSPrintPanel inherits from NSPanel and conforms to NSCoding.

Class Description

NSPrintPanel creates a Print panel used to query the user for information about a print job, such as which pages to print and how many copies, and execute the Print command.

When a **print:** message is sent to an NSView or NSWindow, an NSPrintOperation object is created to control the print operation (see the NSPrintOperation class description for details). By default an NSPrintOperation object uses an NSPrintPanel unless it is sent the **setShowPanels:** message passing NO as the argument. Also, if you subclass NSPrintPanel, send the **setPrintPanel:** message to the NSPrintOperation object passing an instance of your subclass to ensure that it is used as the Print panel for that operation.

However, you rarely need to subclass NSPrintPanel since you can augment its display by adding a custom NSView using the **setAccessoryView:** method. The accessory view is displayed when the user clicks the Options button. (On Mach platforms, the panel is resized to accommodate the NSView that you add.) Note, however, that you don't have to create controls for special printer features. If a printer includes features in the "OpenUI" field of its PostScript Printer Description (PPD) table, these features will appear in the panel. For more information on a printer's PPD table, see the NSPrinter class description.

Typically, you get an NSPrintPanel by invoking the **printPanel** class method. When the class receives a **printPanel** message, it tries to reuse an existing panel rather than create a new one. When a panel is reused, its attributes are reset to the default values so that the effect is the same as returning a new panel. Because a Print panel may be reused, you shouldn't modify the instance returned by **printPanel**, except through the methods listed below. For example, you can set the accessory view, but not the arrangement of the buttons within the panel. If you must modify the Print panel substantially, create and manage your own instance using the **alloc...** and **init...** methods rather than the **printPanel** method.

An application stores printing information in an NSPrintInfo object. When an NSPrintOperation object is created it is given a specific NSPrintInfo object from the application or assigned a default. You can get the current operation by sending the **currentOperation** class method to NSPrintOperation.

Use the **updateFromPrintInfo** method to read the NSPrintInfo object's information into the Print panel. Conversely, the **finalWritePrintInfo** method updates the NSPrintInfo object if the user changes the information on the Print panel. The NSPrintOperation object creates a copy of the NSPrintInfo object, so that **finalWritePrintInfo** actually writes to that copy, not the original.

Method Types

Creating an NSPrintPanel	+ printPanel
Customizing the panel	– accessoryView – setAccessoryView:
Running the panel	– runModal
Communicating with the NSPrintInfo object	– updateFromPrintInfo – finalWritePrintInfo
Updating the panel's display	– pickedButton: – pickedAllPages: – pickedLayoutList:

Class Methods

printPanel

+ (NSPrintPanel *)**printPanel**

Returns a shared NSPrintPanel object or a newly created one if it doesn't already exist.

Instance Methods

accessoryView

– (NSView *)**accessoryView**

Returns the receiver's accessory view (used to customize the receiver).

See also: – **setAccessoryView:**

finalWritePrintInfo

– (void)**finalWritePrintInfo**

Writes the values of the receiver’s printing attributes to the NSPrintInfo object belonging to the current NSPrintOperation.

See also: – **updateFromPrintInfo**, **– currentOperation** (NSPrintOperation)

pickedAllPages:

– (void)**pickedAllPages:**(id)*sender*

This method is for Mach platforms only—it is not defined for other platforms. Invoked when the user chooses the All (Pages) radio button. If the All button is clicked the From and To (Pages) fields are empty, otherwise their default values are set to "first" and "last". Override this method to change these defaults.

See also: – **pickedButton:**, – **pickedLayoutList:**

pickedButton:

– (void)**pickedButton:**(id)*sender*

This method is for Mach platforms only—it is not defined for other platforms. Invoked when the user clicks either the Cancel, Fax, Preview, Print or Save buttons. If a button other than the Cancel button is clicked, then the receiver’s Copies, From (Pages) and To (Pages) fields must contain acceptable values (positive numbers), otherwise the unacceptable entry is selected. If the fields are acceptable the modal loop is stopped. If the Cancel button is selected the modal loop is stopped regardless of the field values.

See also: – **pickedAllPages:**, – **pickedLayoutList:**, – **runModal**

pickedLayoutList:

– (void)**pickedLayoutList:**(id)*sender*

This method is for Mach platforms only—it is not defined for other platforms. Invoked when the user chooses a new layout to update the receiver.

See also: – **pickedAllPages:**, – **pickedButton:**

runModal

– (int)**runModal**

Displays the receiver and begins the modal loop. Returns NSCancelButton if the user clicks the Cancel button, otherwise returns NSOkButton.

See also: – **pickedButton:**

setAccessoryView:

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

Adds an NSView to the receiver. Invoke this method to add a custom view containing your controls. The accessory view is displayed when the user clicks the Options button. (On Mach platforms, the receiver is automatically resized to accommodate *aView*.) This method can be invoked repeatedly to change the accessory view depending on the situation. If *aView* is **nil**, then the receiver's current accessory view, if any, is removed.

See also: – **accessoryView**

updateFromPrintInfo

– (void)**updateFromPrintInfo**

Reads the receiver's values from the NSPrintInfo object belonging to the current NSPrintOperation, and updates the receiver accordingly.

See also: – **finalWritePrintInfo**, – **currentOperation** (NSPrintOperation)