init
initContent:style:backing:buttonMask:defer:
initContent:style:backing:buttonMask:
defer:screen:

Freeing a Window object free

Computing frame and content rectangles
+ getFrameRect:forContentRect:style:
+ getContentRect:forFrameRect:style:
+ minFrameWidth:forStyle:buttonMask:

Accessing the frame rectangle getFrame:
getFrame:andScreen:
setFrameUsingName:
saveFrameUsingName:
+ removeFrameUsingName:
setFrameAutosaveName:
frameAutosaveName
setFrameFromString:
saveFrameToString:

Accessing the content view setContentView:
contentView

Querying Window attributes windowNum
buttonMask
style
worksWhenModal

Window graphics setTitle:
setTitleAsFilename:
title
setBackgroundColor:
backgroundColor

setMiniwindowImage:
setMiniwindowTitle:
miniwindowIcon
miniwindowImage
miniwindowTitle

The field editor endEditingFor:
getFieldEditor:for:

Window status makeKeyWindow
makeKeyAndOrderFront:
becomeKeyWindow
isKeyWindow
resignKeyWindow
canBecomeKeyWindow
becomeMainWindow
isMainWindow
resignMainWindow
canBecomeMainWindow

Moving and resizing
moveTo::screen:
moveTopLeftTo::
moveTopLeftTo::screen:
dragFrom::eventNum:
constrainFrameRect:toScreen:
placeWindow:
placeWindow:screen:
placeWindowAndDisplay:
sizeWindow::
setMinSize:
setMaxSize:
getMinSize:
getMaxSize:
resizeFlags
center

Ordering on and off screen makeKeyAndOrderFront:
orderFront:
orderBack:
orderOut:
orderWindow:relativeTo:
orderFrontRegardless
isVisible
setHideOnDeactivate:
doesHideOnDeactivate

Converting coordinates convertBaseToScreen:
convertScreenToBase:

Managing display display
displayIfNeeded
disableDisplay
isDisplayEnabled
reenableDisplay
flushWindow
flushWindowIfNeeded
disableFlushWindow
reenableFlushWindow

setDynamicDepthLimit:
hasDynamicDepthLimit
canStoreColor

Graphics state objects gState

Cursor management addCursorRect:cursor:forView:
removeCursorRect:cursor:forView:
invalidateCursorRectsForView:
disableCursorRects
enableCursorRects
discardCursorRects
resetCursorRects

Handling user actions and events
close
performClose:
miniaturize:
performMiniaturize:
deminiaturize:
setDocEdited:
isDocEdited
windowExposed:
windowMoved:
screenChanged:

Setting the event mask setEventMask:
addToEventMask:
removeFromEventMask:
eventMask

Aiding event handling getMouseLocation:
setTrackingRect:inside:owner:tag:left:right:
discardTrackingRect:
makeFirstResponder:
firstResponder
sendEvent:
rightMouseDown:
commandKey:
tryToPerform:with:
setAvoidsActivation:
avoidsActivation

Dragging registerForDraggedTypes:count:
unregisterDraggedTypes
dragImage:at:offset:event:pasteboard:source:slideBack:

Services and Windows menu support
validRequestorForSendType:andReturnType:
setExcludedFromWindowsMenu:
isExcludedFromWindowsMenu

Assigning a delegate setDelegate:
delegate

Printing printPSCode:
smartPrintPSCode:
faxPSCode:
smartFaxPSCode:
openSpoolFile:
spoolFile:

endHeaderComments
endPrologue
beginSetup
endSetup
beginPage:label:bBox:fonts:
endPage
beginPageSetupRect:placement:
endPageSetup
beginTrailer
endTrailer

Archiving read:

write:
awake

setDepthLimit:, setDynamicDepthLimit:, canStoreColor

setFrameUsingName:, setFrameAutosaveName:

addCursorRect:(const NXRect *)aRect
cursor:anObject
forView:aView

Adds the rectangle specified by aRect to the Window's list of cursor rectangles and returns self. al
the Window's base coordinate system, must lie within the Window's content rectangle. If it doesn
isn't added and nil is returned.

You typically add cursor rectangles to View objects (through View's addCursorRect:cursor: metho
Windows.

addCursorRect:cursor: (View)

(int)addToEventMask:(int)newEvents

Adds newEvents to the Window's current event mask and returns the original event mask. This m
when an object sets up a modal event loop to respond to certain events. The return value should be
Window's original event mask when the modal loop done. See setEventMask: for a list of event m

setEventMask:, eventMask, removeFromEventMask:

(BOOL)avoidsActivation

Returns YES if the Window's application doesn't become active when the user clicks in the Windo
default is NO. Note that clicking on the title bar will always activate the Window's application.

setAvoidsActivation:

awake

You never invoke this method directly it's invoked automatically after the Window has been read f

read:

(NXColor)backgroundColor

Returns the color of the Window's background when the object is displayed on a color screen. The
equivalent of NX_LTGRAY.

setBackgroundColor:, setBackgroundGray:

(int)backingType

Returns the Window's backing type as one of the following constants:

NX_BUFFERED
NX_RETAINED
NX_NONRETAINED

setBackingType:

becomeKeyWindow

You never invoke this method it's invoked automatically when the Window becomes the key wind
becomeKeyWindow to the Window's first responder, and sends windowDidBecomeKey: to the W
respective objects can respond). Returns self.

makeKeyWindow, makeKeyAndOrderFront:

becomeMainWindow

You never invoke this method it's invoked automatically when the Window becomes the main wir
sends windowDidBecomeMain: to the Window's delegate (if the delegate can respond). Returns s

makeKeyWindow, makeKeyAndOrderFront:

beginPage:(int)ordinalNum
label:(const char *)aString
bBox:(const NXRect *)pageRect
fonts:(const char *)fontNames

Writes a PostScript page separator by forwarding the beginPage:... message to the Window's fram
invoke this method directly it's invoked automatically when printing or faxing the Window.

beginPage:labelbBox:fonts: (View)

beginPageSetupRect:(const NXRect *)aRect
placement:(const NXPoint *)location

Writes the start of a PostScript page-setup section by forwarding the beginPageSetupRect:placeme
Window's frame view. You never invoke this method directly it's invoked automatically when pri
Window.

beginPageSetupRect:placement: (View)

beginPrologueBBox:(const NXRect *)boundingBox
creationDate:(const char *)dateCreated
createdBy:(const char *)anApplication

beginPSOutput

Prepares the Window (and the application environment) for printing or faxing by forwarding the b
to the Window's frame view. You never invoke this method directly it's invoked automatically wh
the Window.

beginPSOutput (View)

beginSetup

Writes the start of a PostScript document-setup section by forwarding the beginSetup message to t
view. You never invoke this method directly it's invoked automatically when printing or faxing th

beginSetup (View)

beginTrailer

Writes the start of a PostScript document-trailer section by forwarding the beginTrailer message to
view. You never invoke this method directly it's invoked automatically when printing or faxing th

beginTrailer (View)

(const NXScreen *)bestScreen

Returns a pointer to the deepest screen that the Window is on, or NULL if the Window is currently

screen, colorScreen (Application)

(int)buttonMask

Returns a mask that indicates which buttons appear in the Window's title bar. The return value ma
of these constants:

NX_CLOSEBUTTONMASK
NX_MINIATURIZEBUTTONMASK

The button mask is set when the Window is initialized and is, thereafter, immutable.

initContent:style:backing:buttonMask:defer:screen:

(BOOL)canBecomeKeyWindow

Returns YES if the Window can be made the key window, and NO if it can't. This method is cons
Window tries to become the key window the attempt is thwarted if this method returns NO.

isKeyWindow, makeKeyWindow

(BOOL)canStoreColor

Returns YES if the Window has a depth limit that allows it to store color values, and NO if it doesn

depthLimit, shouldDrawColor (View)

center

Moves the Window to the center of the screen: The Window is placed dead-center horizontally an above center vertically. Such a placement is consider to carry a certain immediacy and importance typically use this method to place a WindowÐmost likely an attention PanelÐwhere the user can' is invoked automatically when a Panel is placed on the screen by Application's runModalFor: meth

close

Removes the Window from the screen. If the Window is set to be freed when it's closed (the defau sent to the object (but note that the message isn't sent until the current event is completed).

Normally, this method is invoked by the Application Kit when the user clicks the Window's close method doesn't cause windowWillClose: to be sent to the Window's delegate (the message is sent the close button). You can induce an invocation of the delegate method by simulating the user's ac performClose: method.

Returns nil.

performClose:, setFreeWhenClosed:

(BOOL)commandKey:(NXEvent *)theEvent

Responds to the Command key-down event passed as theEvent. You never invoke this method dir object, upon receiving a Command key-down event, sends a commandKey: message to each Wind until one of them returns YES (signifying that the event was recognized and handled). The default method returns NOÐinstances of Window can't handle these events. (By contrast, Panels can.)

You can create your own subclass of Window that responds to Command key-down events. A typ implementation of this method passes a performKeyEquivalent: message down the view hierarchy

performKeyEquivalent: (View), commandKey: (Panel)

(BOOL)constrainFrameRect:(NXRect *)theFrame
toScreen:(const NXScreen *)screen

Modifies the rectangle pointed to by theFrame such that its top edge lies on the given screen. If th the rectangle's height is adjusted to bring the bottom edge onto the screen as well. The rectangle's

contentView

Returns the Window's content view, the highest accessible View object in the Window's view hier

setContentView:

convertBaseToScreen:(NXPoint *)aPoint

Converts the point referred to by aPoint from the Window's base coordinate system to the screen c
Returns self.

convertScreenToBase:

convertScreenToBase:(NXPoint *)aPoint

Converts the point referred to by aPoint from the screen coordinate system to the Window's base c
Returns self.

convertBaseToScreen:

copyPSCodeInside:(const NXRect *)rect to:(NXStream *)stream

Generates PostScript code, in the manner of printPSCode:, for all the Views located inside the rect
Window. The rectangle is specified in the Window's base coordinates. The PostScript code is wri

Returns self (unless an exception is raised).

printPSCode:, faxPSCode:

counterpart

Returns the Window's miniwindow or, if this Window is a miniwindow, the Window that it repres
Window's counterpart directly a corresponding miniwindow is created automatically the first time
miniaturized. If the Window has not yet been miniaturized, this method will return nil.

setMiniwindowImage:, setMiniwindowTitle:

delegate

Returns the Window's delegate, or nil if it doesn't have one.

setDelegate:

deminiaturize:sender

(NXWindowDepth)depthLimit

Returns the depth limit of the Window as one of the following values:

NX_DefaultDepth
NX_TwoBitGrayDepth
NX_EightBitGrayDepth
NX_TwelveBitRGBDepth
NX_TwentyFourBitRGBDepth

If the return value is NX_DefaultDepth, you can find out the actual depth limit by sending the Win
defaultDepthLimit message.

setDepthLimit:, setDynamicDepthLimit:

disableCursorRects

Disables all cursor rectangle management within the Window. Typically this method is used when
special cursor manipulation, and you don't want the Application Kit interfering. Returns self.

enableCursorRects

disableDisplay

Disables View's display methods, thus preventing the Views in the Window's view hierarchy from
however, that this doesn't disable Window's display method). This permits you to alter or update
displaying them again.

Displaying should be disabled only temporarily. Each disableDisplay message should be paired w
reenableDisplay message. Pairs of these messages can be nested drawing won't be reenabled until
reenableDisplay message is sent or until a display message is sent to the Window.

Returns self.

reenableDisplay, isDisplayEnabled, display, display::: (View)

disableFlushWindow

Disables the flushWindow method for the Window. If the Window is a buffered window, drawing
flushed to the screen by the display methods defined in the View class. This permits several View
the results are shown to the user.

Flushing should be disabled only temporarily, while the Window's display is being updated. Each
message should be paired with a subsequent reenableFlushWindow message. Message pairs can b
be reenabled until the last (unnested) reenableFlushWindow message is sent.

Returns self.

reenableFlushWindow, flushWindow, disableDisplay

discardCursorRects

Removes the tracking rectangle identified by trackNum and returns self. The tag was assigned wh
rectangle was created.

setTrackingRect:inside:owner:tag:left:right:

display

Passes a display message down the Window's view hierarchy, thus redrawing all Views within the
border, resize bar, and title bar. If displaying is disabled for the Window, this method reenables it.

display (View), disableDisplay, displayIfNeeded

displayBorder

Redraws the Window's border, title bar, and resize bar, and returns self. You rarely need to invok
Window's border is automatically displayed when any of the elements therein are changedÐwhen
or its title is changed, for example.

display

displayIfNeeded

Sends a displayIfNeeded message down the Window's view hierarchy, thus redrawing all Views th
displayed, including the Window's border, title bar, and resize bar. This method is useful when yo
displaying in the Window, modify some number of Views, and then display only the ones that wer
this method, unlike display, doesn't reenable display if it's currently disabled. Returns self.

display, displayIfNeeded (View), setNeedsDisplay: (View), update (View)

(BOOL)doesHideOnDeactivate

Returns YES if the Window will be removed from the screen when its application is deactivated, a
on-screen.

setHideOnDeactivate:

dragFrom:(float)x
:(float)y
eventNum:(int)num

Lets the user drag a Window from a location other than the title bar.

mouseDown:(NXEvent *)theEvent

dragImage:anImage
at:(NXPoint *)location
offset:(NXPoint *)initialOffset
event:(NXEvent *)event
pasteboard:(Pasteboard *)pboard
source:sourceObject
slideBack:(BOOL)slideFlag

Instigates an image-dragging session. You never invoke this method directly from your applicatio invoked from within a View's implementation of the mouseDown: method. Furthermore, View al dragImage:... method you typically instigate an image-dragging session by sending this message to Window. The two methods are identical except for the interpretation of the location argument: In implementation, location is taken in the base coordinate system. See the description of this metho the meanings of the other arguments.

dragImage:at:offset:event:pasteboard:source:slideBack: (View)

enableCursorRects

Reenables cursor rectangle management. Returns self.

disableCursorRects

endEditingFor:anObject

Prepares the Window's field editor for a new editing assignment and returns self. The argument is default implementation.

If the field editor is the first responder, it resigns that status, passing it to the Window (even if the f resign). This forces a textDidEnd:endChar: message to be sent to the field editor's delegate. The f removed from the view hierarchy and its delegate is set to nil.

To conditionally end editing, first try to make the Window the first responder:

getFieldEditor:for:

endHeaderComments

Writes the end of a PostScript comment section by forwarding the endHeaderComments message t view. You never invoke this method directly it's invoked automatically when printing or faxing th

endHeaderComments (View)

endPage

You never invoke this method directly it's invoked automatically when printing or faxing the Win

endPageSetup (View)

endPrologue

Writes the end of a PostScript prolog section by forwarding the endPrologue message to the Windo
never invoke this method directly it's invoked automatically when printing or faxing the Window.

endPrologue (View)

endPSOutput

Declares that printing or faxing is finished by forwarding the endPSOutput to the Window's frame
invoke this method directly it's invoked automatically when printing or faxing the Window.

endPSOutput (View)

endSetup

Writes the end of a PostScript document-setup section by forwarding the endSetup message to the
You never invoke this method directly it's invoked automatically when printing or faxing the Win

endSetup (View)

endTrailer

Writes the end of a PostScript document-trailer section by forwarding the endTrailer message to th
view. You never invoke this method directly it's invoked automatically when printing or faxing th

endTrailer (View)

(int)eventMask

Returns the current event mask for the Window. See setEventMask: for a list of the possible conte

setEventMask:, addToEventMask:, removeFromEventMask:

faxPSCode:sender

Prints the Window (all the Views in its view hierarchy including the frame view) to a fax modem.
indicates that there were errors in generating the PostScript code or that the user canceled the job.

In the current user interface, faxing is initiated from within the Print panel. However, with this me
users with an independent control for faxing a Window.

Returns the Window's first responder.

makeFirstResponder:, acceptsFirstResponder (Responder)

flushWindow

if the Window is buffered and flushing hasn't been disabled by disableFlushWindow, this flushes t
the screen. This method is automatically invoked when you send a display message to a Window o
has no effect if the display is being directed to a printer or other device, rather than to the screen. I

display:: (View), disableFlushWindow

flushWindowIfNeeded

Flushes the Window's off-screen buffer to the screen, provided that:

·The Window is a buffered window
·Flushing isn't currently disabled
·Some previous flushWindow messages had no effect because flushing was disabled

You should use this method, rather than flushWindow, to flush a Window after flushing has been r

flushWindow, disableFlushWindow, reenableFlushWindow

(const char *)frameAutosaveName

Returns the name that's used to automatically save the Window's frame rectangle data in the defau
through setFrameAutosaveName:. If the Window has an autosave name, it's frame data is written
the frame rectangle changes.

setFrameAutosaveName:

free

Deallocates memory for the Window object and all that it surveys. This includes the Views in its
instance variables (including the field editor), and the Window Server window device that it's asso

getFieldEditor:(BOOL)flag for:anObject

Returns the field editor, the Window's communal Text object. The field editor is provided as a co
used however your application sees fit. Typically, the field editor is used by simple text-bearing ob
TextField object uses its Window's field editor to display and manipulate text. The field editor ca
number of objects and so its state may be constantly changing. Therefore, it shouldn't be used to d
demands sophisticated Text object preparation (for this you should create a dedicated Text object).

A freshly created Window doesn't have a field editor the only way to create a field editor is to invo
flag value of YES. After a field editor has been created for a Window, the flag argument is ignore

endEditingFor:

getFrame:(NXRect *)theRect

Returns the Window's frame rectangle by reference in theRect and returns self. The frame rectang
in the screen coordinate system.

getFrame:andScreen:

getFrame:(NXRect *)theRect andScreen:(const NXScreen **)theScreen

Copies the Window's frame rectangle into the structure referred to by theRect. A pointer to the sc
is located is provided in the variable referred to by theScreen. The frame rectangle is specified rel
corner of the screen. However, if theScreen is NULL, the frame rectangle is specified in absolute
the origin of the screen coordinate system). Returns self.

getFrame:

getMaxSize:(NXSize *)aSize

Returns, by reference in aSize, an NXSize structure that gives the maximum size to which the Win
sized by the user or by the setFrame:... methods. Note that this constraint doesn't apply to sizeWi
placeWindow:... methods.

setMaxSize:, setMinSize:, getMinSize:

getMinSize:(NXSize *)aSize

Returns, by reference in aSize, an NXSize structure that gives the minimum size to which the Win
sized by the user or by the setFrame:... methods. Note that this constraint doesn't apply to sizeWi
placeWindow:... methods.

setMinSize:, setMaxSize:, getMaxSize:

getMouseLocation:(NXPoint *)thePoint

Returns, by reference in thePoint, the current location of the mouse reckoned in the Window's bas
Returns self.

currentEvent (Application)

(BOOL)getRect:(NXRect *)theRect forPage:(int)page

Implemented by subclasses to provide the rectangle to be printed for page number page. A Windo
forPage: messages when it's being printed (or faxed) if its knowsPagesFirst:last: method returns Y

(int)gState

Returns the PostScript graphics state object associated with the Window.

(BOOL)hasDynamicDepthLimit

Returns YES if the Window's depth limit can change to match the depth of the screen it's on, and

setDynamicDepthLimit:

(float)heightAdjustLimit

Returns the fraction of a page that can be pushed onto the next page to prevent items from being cu
applies to vertical pagination. By default, it's 0.2.

You never invoke this method directly it's invoked during automatic pagination when printing (or
However, you can override it to return a different value. The value returned should lie between 0.0

widthAdjustLimit

init

Initializes the receiver, a newly allocated Window object, by passing default values to the initCont
buttonMask:defer: method. The initialized object is a plain, buffered window, and has a default fra
self.

initContent:style:backing:buttonMask:defer:

initContent:(const NXRect *)contentRect
style:(int)aStyle
backing:(int)backingType
buttonMask:(int)mask
defer:(BOOL)flag

Initializes the Window object and returns self. This method is the designated initializer for the Wi

The first argument, contentRect, specifies the location and size of the Window's content area in sc
NULL pointer is passed for this argument, a default rectangle is used.

The second argument, aStyle, specifies the Window's style. It can be:

NX_PLAINSTYLE
NX_TITLEDSTYLE
NX_RESIZEBARSTYLE
NX_MENUSTYLE
NX_MINIWINDOWSTYLE
NX_MINIWORLDSTYLE
NX_TOKENSTYLE

The fourth argument, mask, specifies whether the Window's title bar will sport a close or resize bu
mask by joining (with the bitwise OR operator) the individual masks for the buttons:

NX_CLOSEBUTTONMASK
NX_MINIATURIZEBUTTONMASK

The fifth argument, flag, determines whether the Window Server will create a window device for t
immediately. If flag is YES, it will defer creating the window until the Window is ordered on-scre
messages sent to the Window or its Views will be postponed until the window is created, just befo
screen. Deferring the creation of the window improves launch time and minimizes the virtual mem

The Window creates an instance of View to be its default content view. You can replace it with yo
the setContentView: method.

orderFront:, setTitle:, setOneShot:

initContent:(const NXRect *)contentRect
style:(int)aStyle
backing:(int)bufferingType
buttonMask:(int)mask
defer:(BOOL)flag
screen:(const NXScreen *)aScreen

Initializes the Window object and returns self. This method is equivalent to initContent:style:back
except that the content rectangle is specified relative to the lower left corner of aScreen.

If aScreen is NULL, the content rectangle is interpreted relative to the lower left corner of the mai
screen is the one that contains the current key window, or, if there is no key window, the one that c
If there's neither a key window nor a main menu (if there's no active application), the main screen
origin of the screen coordinate system is located.

initContent:style:backing:buttonMask:defer:

invalidateCursorRectsForView:aView

Marks the Window as having invalid cursor rectangles. If the Window is the key window, the App
send it a resetCursorRects message to have it fix its cursor rectangles before getting the next event
the key window, it will receive the message when it next becomes the key window. Returns self.

resetCursorRects

(BOOL)isDisplayEnabled

Returns YES if the display mechanism is currently disabled (because of a previous disableDisplay
isn't.

disableDisplay, reenableDisplay, display::: (View)

(BOOL)isDocEdited

Returns YES if the Window's document has been edited, otherwise returns NO.

(BOOL)isFlushWindowDisabled

Returns YES if the Window's flushing ability has been disabled otherwise returns NO.

disableFlushWindow, reenableFlushWindow

(BOOL)isKeyWindow

Returns YES if the Window is the key window for the application, and NO if it isn't.

isMainWindow

(BOOL)isMainWindow

Returns YES if the Window is the main window for the application, and NO if it isn't.

isKeyWindow

(BOOL)isOneShot

Returns YES if the window device that the Window manages is freed when it's removed from the
not. The default is NO.

setOneShot:

(BOOL)isVisible

Returns YES if the Window is on-screen (even if it's obscured by other Windows).

getVisibleRect: (View)

(BOOL)knowsPagesFirst:(int *)firstPageNum last:(int *)lastPageNum

Implemented by subclasses to indicate whether the Window knows where its own pages lie. This
when printing (or faxing) the Window. Although it can be implemented in a Window subclass, it
program code.

If this method returns YES, the Window will receive getRect:forPage: messages querying it for the
corresponding to specific pages. If it returns NO, pagination will be done automatically. By defau

Just before this method is invoked, the first page to be printed is set to 1 and the last page to be pri
maximum integer size. An implementation of this method can set firstPageNum to a different initi
chapter may start on page 40), even if it returns NO. If it returns YES, lastPageNum can be set to
it doesn't reset lastPageNum, the subclass implementation of getRect:forPage: must be able to sign
asked for beyond what is available in the document.

getRect:forPage:, printPSCode:

The Application Kit uses this method to alter the first responder in response to mouse-down events
explicitly set the first responder from within your program. aResponder should be a Responder ob
View in the Window's view hierarchy.

If successful in making aResponder the first responder, this method returns self. If not (if the old f
to resign), it returns nil.

becomeFirstResponder (Responder), resignFirstResponder (Responder)

makeKeyAndOrderFront:sender

Moves the Window to the front of the screen list (within its tier) and makes it the key window. Th
in action message. Returns self.

orderFront:, orderBack:, orderOut:, orderWindow:relativeTo:

makeKeyWindow

Makes the Window object the key window, and returns self.

becomeKeyWindow, isKeyWindow

miniaturize:sender

Removes the Window from the screen list and displays its miniwindow counterpart on-screen. If t
have a miniwindow counterpart, one is created.

A miniaturize: message is generated when the user clicks the miniaturize button in the Window's t
has a sender argument so that it can be used in an action message from a Control. It ignores this ar

deminiaturize:

(const char *)miniwindowIcon

Returns the name of the icon that's displayed in the Window's miniwindow.

setMiniwindowIcon:

(NXImage *)miniwindowImage

Returns the NXImage object that's displayed in the Window's miniwindow.

setMiniwindowImage:

(const char *)miniwindowTitle

Returns the title that's displayed in the Window's miniwindow.

moveTo:(NXCoord)x :(NXCoord)y screen:(const NXScreen *)aScreen

Repositions the Window so that its lower left corner lies at (x, y) relative to a coordinate origin at t
aScreen. If aScreen is NULL, this method is the same as moveTo::. Returns self.

moveTopLeftTo:(NXCoord)x :(NXCoord)y

Moves the Window by the top left corner of its frame rectangle. The arguments are taken in the sc
Returns self.

dragFrom::eventNum:, moveTo::

moveTopLeftTo:(NXCoord)x :(NXCoord)y screen:(const NXScreen *)aScreen

Repositions the Window so that its top left corner lies at (x, y) relative to a coordinate origin at the
aScreen. If aScreen is NULL, this method is the same as moveTopLeftTo::. Returns self.

moveTo::

openSpoolFile:(char *)filename

Opens the filename file for print spooling. This method is invoked when printing (or faxing) the W
used in program code. However, you can override it to modify its behavior.

If filename is NULL or empty, PostScript code for the Window will be sent directly to the printing
opening a file. (However, if the Window is being previewed or saved, a default file is opened in /t

If a filename is provided, the file is opened. The printing machinery will then write the PostScript
file will be printed using lpr.

This method opens a Display PostScript context that will write to the spool file, and sets the contex
PrintInfo object to this new context. It returns nil if the file can't be opened.

printPSCode:

orderBack:sender

Moves the Window to the back of its tier in the screen list. It may also change the key window and
Returns self.

orderFront:, orderOut:, orderWindow:relativeTo:, makeKeyAndOrderFront:

orderFront:sender

moved in front of the key window unless the Window and the key window are in the same applicat
need to invoke this method it's designed to be used when applications are cooperating such that an
(with the key window) is using another application to display data.

orderFront:

orderOut:sender

Takes the Window out of the screen list. It may also change the key window and main window. F

orderFront:, orderBack:, orderWindow:relativeTo:

orderWindow:(int)place relativeTo:(int)otherWin

Repositions the Window's window device in the Window Server's screen list. place can be one of

NX_ABOVE
NX_BELOW
NX_OUT

If it's NX_OUT, the window is removed from the screen list and otherWin is ignored. If it's NX_
NX_BELOW, otherWin is the window number of the window that the receiving Window is to be p
If otherWin is 0, the receiving Window will be placed above or below all other windows in its tier.

orderFront:, orderBack:, orderOut:, makeKeyAndOrderFront:

performClose:sender

Simulates the user clicking the close button by momentarily highlighting the button and then closi
Window's delegate or the Window itself implements windowWillClose:, then that message is sent
the argument (only one such message is sent if both the delegate and the Window implement the m
delegate will receive the message).

If the Window doesn't have a close button, then the method calls NXBeep(). Returns self.

performClick: (Button), close, performMiniaturize:

performMiniaturize:sender

Simulates the user clicking the miniaturize button by momentarily highlighting the button then mi
If the Window doesn't have a miniaturize button, then this method calls NXBeep(). Returns self.

performClick: (Button), miniaturize:, performClose:

placePrintRect:(const NXRect *)aRect offset:(NXPoint *)location

Determines the location of the rectangle being printed on the physical page. You never invoke this
automatically invoked when the Window is printed or faxed. However, you can override it to char
the rectangle.

placeWindow:(const NXRect *)frameRect

Resizes and moves the Window. frameRect specifies the Window's new frame rectangle in screen
Window's frame viewÐbut none of its other ViewsÐis automatically redisplayed at its new size a
self.

sizeWindow::, moveTo::, placeWindowAndDisplay:

placeWindow:(const NXRect *)frameRect screen:(const NXScreen *)aScreen

This is the same as placeWindow:, except that the frame rectangle is specified relative to a coordin
left corner of aScreen. If aScreen is NULL, this method is exactly the same as placeWindow:. Re

placeWindow:, placeWindowAndDisplay:

placeWindowAndDisplay:(const NXRect *)frameRect

This is the same as placeWindow:, except the Window's Views are redisplayed before the Windov
self.

placeWindow:

printPSCode:sender

Prints the Window (all the Views in its view hierarchy including the frame view). A return value o
there were errors in generating the PostScript code or that the user canceled the job.

This method normally brings up the Print panel before actually beginning printing. But if sender i
shouldRunPrintPanel: method, that method will be invoked to first query whether to run the panel.
shouldRunPrintPanel: returns NO, the Print panel won't be displayed, and the Window will be prin
settings of the panel.

smartPrintPSCode:, faxPSCode:, shouldRunPrintPanel: (Object Methods)

read:(NXTypedStream *)stream

Reads the Window and its Views from the typed stream stream.

write:

reenableDisplay

Counters the effect of disableDisplay, reenabling View's display methods. Returns self.

disableDisplay, isDisplayEnabled, display::: (View)

registerForDraggedTypes:(const char *const *)pbTypes count:(int)count

Registers the Pasteboard types that the Window will accept in an image-dragging session. pbTypes array of the types count is the number of elements in the array. Returns self.

Keep in mind that the values in the first argument are Pasteboard types, not file extensions (you ca file extensions). For example, the following registers a Window as accepting files:

unregisterDraggedTypes

removeCursorRect:(const NXRect *)aRect
cursor:anObj
forView:aView

Removes a cursor rectangle from the Window. You never invoke this method it's used by View's cursor: method. To remove a cursor rectangle, use the View method.

removeCursorRect:cursor: (View), resetCursorRects (View)

(int)removeFromEventMask:(int)oldEvents

Removes the event types specified by oldEvents from the Window's event mask, and returns the o

eventMask, setEventMask:, addToEventMask:

resetCursorRects

Removes all existing cursor rectangles from the Window, then recreates the cursor rectangles by se resetCursorRects message to every View in the Window's view hierarchy. Returns self.

This method is typically invoked by the Application object when it detects that the key window's c invalid. In program code, it's more efficient to invoke invalidateCursorRectsForView:, rather than invalid cursor rectangles.

invalidateCursorRectsForView:, resetCursorRects (View)

resignKeyWindow

You never invoke this method it's invoked automatically when the Window resigns key window st sends resignKeyWindow to the Window's first responder, and sends windowDidResignKey: to the the respective objects can respond). Returns self.

becomeKeyWindow

resignMainWindow

event that initiated the resizing session. The integer encodes, as a mask, information such as whicl
was held down when the event occurred. The flags are listed in dpsclient/event.h. Because of its l
method should only be invoked from within an implementation of the delegate methods windowW
windowDidResize:.

rightMouseDown:(NXEvent *)theEvent

Responds to uncaught right mouse-down events by forwarding this message to the Application obj
mouse-down event in a window causes the main menu to pop up under the cursor. Returns the val
Application object.

rightMouseDown: (Application)

(void)saveFrameToString:(char *)string

Saves the Window's frame rectangle data as a NULL-terminated ASCII string to the buffer pointe
string can be stored as you see fit and used later to set the dimensions of a Window through the set
method. You should use the constant NX_MAXFRAMESTRINGLENGTH to allocate the buffer.

setFrameFromString:, saveFrameUsingName:

(void)saveFrameUsingName:(const char *)name

Saves the Window's frame rectangle as a system default. With the companion method setFrameU
save and reset a Window's frame over various launchings of an application. The default is owned
under the name

ªWindow Frame nameº

setFrameUsingName:, saveFrameToString:

(const NXScreen *)screen

Returns a pointer to the screen that the Window is on. If the Window is partly on one screen and p
screen where most of it lies is the one returned.

bestScreen

screenChanged:(NXEvent *)theEvent

Invoked when the user releases the Window, having moved all or part of it to a different screen. T
delegate a windowDidChangeScreen: message (if the delegate can respond) and returns self.

If the Window has a dynamic depth limit, this method will make sure that the depth limit matches
Window is on more than one screen, its depth limit will be adjusted to match the deepest screen it'

bestScreen

Establishes whether the Window's application will become the active application when the user cl
content area. If flag is YES, the application won't become active if flag is NO, it will. The defaul
clicking on the title bar will always activate the Window's application.

avoidsActivation

setBackgroundColor:(NXColor)color

Sets the color that fills the Window's content area when the Window is displayed on a color screen

backgroundColor

setBackgroundGray:(float)value

Sets the shade of gray that fills the Window's content area when the Window is displayed on a mo
should lie in the range 0.0 (black) to 1.0 (white). Returns self.

backgroundGray, setBackgroundColor:

setBackingType:(int)backing

Sets the type of backing used by the Window's window device and returns self. This method can
buffered Window to retained or vice versa you can't change the backing type of a nonretained Wi
is generated if you attempt to do so).

backingType

setContentView:aView

Makes aView the Window's content view the previous content view is removed from the Window
returned by this method. aView is resized to fit precisely within the content area of the Window.
content view's coordinate system, but you can't alter its size or location directly.

contentView