opaqueAncestor
removeFromSuperview
replaceSubview:with:
subviews
superview
window
windowChanged:

Modifying the frame rectangle frameAngle
getFrame:
moveBy::
moveTo::
rotateBy:
rotateTo:
setFrame:
sizeBy::
sizeTo::

Modifying the coordinate system
boundsAngle
drawInSuperview
getBounds:
isFlipped
isRotatedFromBase
isRotatedOrScaledFromBase
rotate:
scale::
setDrawOrigin::
setDrawRotation:
setDrawSize::
setFlipped:
translate::

Converting coordinates centerScanRect:
convertPoint:fromView:
convertPoint:toView:
convertPointFromSuperview:
convertPointToSuperview:
convertRect:fromView:
convertRect:toView:
convertRectFromSuperview:
convertRectToSuperview:
convertSize:fromView:
convertSize:toView:

Notifying ancestor Views descendantFlipped:
descendantFrameChanged:
notifyAncestorWhenFrameChanged:
notifyWhenFlipped:
suspendNotifyAncestorWhenFrameChanged:

Resizing subviews resizeSubviews:
setAutoresizeSubviews:
setAutosizing:
autosizing
superviewSizeChanged:

Graphics state objects allocateGState
freeGState

unlockFocus

Displaying canDraw
display
display::
display:::
displayFromOpaqueAncestor:::
displayIfNeeded
drawSelf::
getVisibleRect:
isAutodisplay
setAutodisplay:
isOpaque
setOpaque:
needsDisplay
setNeedsDisplay:
shouldDrawColor
update

Scrolling adjustScroll:
autoscroll:
calcUpdateRects::::
invalidate::
scrollPoint:
scrollRect:by:
scrollRectToVisible:

Managing the cursor addCursorRect:cursor:
discardCursorRects
removeCursorRect:cursor:
resetCursorRects

Assigning a tag findViewWithTag:
tag

Aiding event handling acceptsFirstMouse
hitTest:
mouse:inRect:
performKeyEquivalent:
shouldDelayWindowOrderingForEvent:

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

Printing printPSCode:
faxPSCode:
faxPSCode:toList:numberList:sendAt:wantsCover:
wantsNotify:wantsHires:faxName:
copyPSCodeInside:to:
writePSCodeInside:to:
openSpoolFile:
spoolFile:
canPrintRIB

Setting up pages
getRect:forPage:
placePrintRect:offset:
heightAdjustLimit
widthAdjustLimit

beginPageSetupRect:placement:
drawSheetBorder::
drawPageBorder::
addToPageSetup
endPageSetup
endPage
beginTrailer
endTrailer
endPSOutput

Archiving awake

read:
write:

(BOOL)acceptsFirstMouse

This returns YES if an initial mouse-down event in the ViewÐan event that causes the View's Wi
key windowÐis sent to the View (through a mouseDown: message). If only those mouse-downs t
View's Window is already key are sent, this returns NO (the default). The only way to change the
implement this method in a View subclass.

addCursorRect:(const NXRect *)aRect cursor:anNXCursor

Creates a cursor rectangle, an area within the View that has its own cursor: When the user moves
rectangle specified by aRect, the cursor object that the mouse controls changes to anNXCursor, wh
NXCursor object. The rectangle is given in the View's coordinate system however, the rectangle i
clipped to the View's frameÐit's possible to create a cursor rectangle that extends beyond the Vie
note that cursor rectangles don't work well in rotated Views.

You never invoke this method directly from your application. It should only be used as part of the
resetCursorRects method.

Returns self.

resetCursorRects

addSubview:aView

Adds aView to the View's list of subviews such that new subview will be displayed on top of its si
View is also made aView's next responder. Returns aView (or nil if it isn't a View).

addSubview::relativeTo:, subviews, removeFromSuperview, setNextResponder: (Responder)

addSubview:aView
:(int)place
relativeTo:otherView

Allows applications to add a scaling operator to the PostScript code generated when printing if you operator, this is the correct place to do so. This method is invoked by printPSCode: and faxPSCod method simply returns self this method can be overridden by applications that implement their own

beginPageSetupRect:placement:

adjustPageHeightNew:(float *)newBottom
top:(float)oldTop
bottom:(float)oldBottom
limit:(float)bottomLimit

Adjusts page height for automatic pagination when printing the View. This method is invoked by faxPSCode: to set newBottom, which will be the new bottom of the strip to be printed for the curre oldBottom are the current values for the horizontal strip to be printed. bottomLimit is the topmost be set to. If this limit is exceeded, newBottom is set to oldBottom. By default this method tries to in two. All parameters are in the View's own coordinate system. Returns self.

adjustPageWidthNew:(float *)newRight
left:(float)oldLeft
right:(float)oldRight
limit:(float)rightLimit

Adjusts page width for automatic pagination when printing the View. This method is invoked by p faxPSCode: to set newRight, which will be the new right edge of the strip to be printed for the curr oldRight are the current values for the vertical strip to be printed. rightLimit is the leftmost value If this limit is exceeded, newRight is set to oldRight. By default this method tries to not let the Vie parameters are in the View's own coordinate system. Returns self.

adjustScroll:(NXRect *)newVisible

Allows you to correct the scroll position of a document. This method is invoked by a ClipView im scrolling its document view. You may want to override it to provide specific scrolling behavior. visible rectangle after the scroll. You might use this for scrolling through a table as in a spreadshe newVisible->origin such that the scroll would fall on column or row boundaries. Returns self.

allocateGState

Explicitly tells the View to allocate a graphics state object. Graphics state objects are Display Post contain the entire state of the graphics environment. They are used by the Application Kit as a cac save PostScript code used for focusing, purely as a performance optimization. You can allocate a Views that will be focused on repeatedly, but you should exercise some discretion as they can take memory. The graphics state object will be freed automatically when the View is freed. Returns se

freeGState

(unsigned int)autosizing

Returns the View's autosizing mask. The mask is used to determine how the View is automaticall superview is resized. For the mask to have an effect, the superview must be set to resize its subvie the setAutoresizeSubviews: method. The autosizing masks are listed under the setAutosizing: met

setAutosizing:, setAutoresizeSubviews:

awake

Invoked after unarchiving to allow the View to perform additional initialization. Returns self.

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

Writes a conforming Postscript page separator. This method is invoked by printPSCode: and faxP

ordinalNum specifies the page's position in the document's page sequence (from 1 through n for a

aString is a string that contains no white space characters. It identifies the page according to the do numbering scheme. If aString is NULL, the ASCII equivalent of ordinalNum is used.

pageRect is the rectangle enclosing all the drawing on the page about to be printed in the default P system of the page. If pageRect is NULL, ª(atend)º is output instead of a description of the bound bounding box is output at the end of the page.

fontNames is a string containing the names of the fonts used in this page. Each name should be se the fonts used are unknown before the page is printed, fontNames can be NULL. They will then b the end of the page description. Returns self.

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

Writes the page setup section for a page. This method is invoked by printPSCode: and faxPSCode comments for the page have been written. It outputs a PostScript save, and generates the initial co to set this View up for printing the aRect rectangle within the View. This method does a lockFocu must be balanced in endPage by an unlockFocus. The save output here should be balanced by a Po endPage. aRect is the rectangle in the View's coordinates that is being printed. location is the offs of the rectangle on the physical page. Returns self.

printPSCode, endPage, lockFocus, addToPageSetup

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

dateCreated is an ASCII string containing a human readable date. If dateCreated is NULL the curr

anApplication is a string containing the name of the document creator. If anApplication is NULL t
by Application's appName method is used.

fontNames is a string holding the names of the fonts used in the document. Names should be separ
fonts used are unknown before the document is printed, fontNames should be NULL. In this case
referenced by a findFont is written in the trailer.

user is a string containing the name of the person the document is being printed for. If NULL the l
is used.

numPages specifies the number of pages in the document. If unknown at the beginning of printing
have a value of -1. In this case the pages are counted as they are generated and the resulting count

aTitle is a string specifying the title of the document. If aTitle is NULL, then the title of the View'
the Window has no title, ªUntitledº is output. Returns self.

appName (Application)

beginPSOutput

Performs various initializations before actual PostScript generation begins. This method makes the
context stored in the Application object's global PrintInfo object into the current context. This has
redirecting all PostScript output from the Window Server to the spool file or printer. This method
printPSCode: and faxPSCode: just before any PostScript is generated. Returns self.

beginSetup

Writes the beginning of the document setup section, which begins with a %%BeginSetup commen
%%PaperSize comment declaring the type of paper being used. This method is invoked by printPS
at the start of the setup section of the document, which occurs after the prologue of the document h
before any pages are written. This section of the output is intended for device setup or general init
self.

beginTrailer

Writes the start of a conforming PostScript trailer. This method is invoked by printPSCode: and fa
after all pages have been written. Returns self.

(float)boundsAngle

Returns the angle of the View's bounds rectangle relative to its frame rectangle. If the View's coo
rotated, this angle will be the accumulation of all rotate: messages otherwise, it will be 0.0.

rotate:, setDrawRotation:

(BOOL)calcUpdateRects:(NXRect *)rects

generated. If rectCount is 1, the area that needs to be updated is in rects[0]. If rectCount is 3, the a
updated are in rects[1] and rects[2], and rects[0] is the same as enclRect.

Returns YES if any update rectangles were generated (in other words, if rectCount is greater than z
NO.

scrollRect:by:, NXIntersectionRect()

(BOOL)canDraw

Informs you of whether drawing will have any result. You only need to send this message when y
but are not invoking one of the display methods. You should not draw or send the lockFocus: mes
This method returns YES if your View has a Window object, your View's Window object has a co
the Window Server, and your Window object is enabled for display otherwise it returns NO.

isDisplayEnabled (Window)

(BOOL)canPrintRIB

Indicates whether the View can print RIB files.

centerScanRect:(NXRect *)aRect

Converts the corners of a rectangle to lie on the center of device pixels. This is useful in compensa
overscanning when the coordinate system has been scaled. This routine converts the given rectang
coordinates, adjusts the rectangle to lie in the center of the pixels, and converts the resulting rectan
coordinate system. Returns self.

clipToFrame:(const NXRect *)frameRect

Allows the View to do arbitrary clipping during focusing. This method is invoked from within the
clipping is required. If you override this method, you must use frameRect rather than the View's f
because the origins may not be the same due to focusing. The following example demonstrates cli
circular region:

this method converts from window base coordinates. Both aView and the receiving View must be
Window. Returns self.

convertPoint:(NXPoint *)aPoint toView:aView

Converts a point from the receiving View's coordinate system to the coordinate system of aView.
this method converts to window base coordinates. Both aView and the receiving View must belon
Returns self.

convertPointFromSuperview:(NXPoint *)aPoint

Converts a point from the coordinate system of the receiving View's superview to the coordinate s
View. Returns self.

convertRectFromSuperview:, convertPointToSuperview:

convertPointToSuperview:(NXPoint *)aPoint

Converts a point from the View's coordinate system to that of its superview. Returns self.

convertPointFromSuperview:, convertPoint:fromView:

convertRect:(NXRect *)aRect fromView:aView

Converts aRect from aView's coordinate system to the coordinate system of the receiving View. F
receiving View must belong to the same Window. Returns self.

convertRect:(NXRect *)aRect toView:aView

Converts aRect from the receiving View's coordinate system to the coordinate system of aView. F
receiving View must belong to the same Window. Returns self.

convertRectFromSuperview:(NXRect *)aRect

Converts aRect from the coordinate system of the receiving View's superview to the coordinate sy
View. Returns self.

convertRectToSuperview:

convertRectToSuperview:(NXRect *)aRect

Converts aRect from the receiving View's coordinate system to the coordinate system of its superv

convertSize:(NXSize *)aSize toView:aView

Converts aSize from the receiving View's coordinate system to the coordinate system of aView. B
receiving View must belong to the same Window. Returns self.

convertSize:fromView:

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

Generates PostScript code for the View and all its subviews for the area indicated by rect. The Pos
to the NXStream stream. Returns self, assuming no exception is raised in the generation of PostSc
exception is raised, control is given to the appropriate error handler, and this method does not retur

descendantFlipped:sender

Notifies the receiving View that sender, a View below the receiving View in the view hierarchy, ha
flipped. A descendantFlipped: message is sent from the setFlipped: method if a notifyWhenFlippe
previously sent to sender.

View's default implementation of this method simply passes the message to the receiving View's s
the superview's return value. View subclasses should override this method to respond to the messa
Application Kit, ClipView overrides this method to keep its coordinate system aligned with its doc

notifyWhenFlipped:, setFlipped:, descendantFlipped: (ClipView)

descendantFrameChanged:sender

Notifies the receiving View that sender, a View below the receiving View in the view hierarchy, w
descendantFrameChanged: message is sent from the sizeTo:: and moveTo:: methods if a
notifyAncestorWhenFrameChanged:YES message was previously sent to sender.

View's default implementation of this method simply passes the message to the receiving View's s
the superview's return value. View subclasses should override this method to respond to the messa
Application Kit, the ClipView class overrides this method to notify the ScrollView to reset scroller
document view's frame is changed.

notifyAncestorWhenFrameChanged:, sizeTo::, moveTo::

discardCursorRects

Removes the View's cursor rectangles. You never invoke this method directly it's invoked automa
View's cursor rectangles are reset. Returns self.

resetCursorRects, discardCursorRects (Window)

display:(const NXRect *)rects :(int)rectCount

Displays the View and its subviews. The rectangles are specified in the receiving View's coordina self. This method is equivalent to:

display:::, drawSelf::

display:(const NXRect *)rects
:(int)rectCount
:(BOOL)clipFlag

Displays the View and its subviews by invoking the lockFocus, drawSelf::, and unlockFocus meth drawing rectangles in the receiving View's coordinate system they're used to restrict what is displa number of valid rectangles in rects (0, 1, or 3).

If rectCount is 3, then rects[0] should contain the smallest rectangle that completely encloses rects rectangles that actually specify the regions to be displayed.

If rectCount is 1, rects[0] should specify the region to be displayed.

If rectCount is 0 or rects is NULL, the View's visible rectangle is substituted for rects[0] and a val rectCount.

In any case, the rectangles in rects are intersected against the visible rectangle.

This method doesn't display a subview unless it falls at least partially inside rects[0] if rectCount i rects[1] or rects[2] if rectCount is 3. When this method is applied recursively to each subview, the translated to the subview's coordinate system and intersected with its bounds rectangle to produce rectCount are then passed as arguments to each View's drawSelf:: method.

If clipFlag is YES, this method clips to the drawing rectangles. Clipping isn't done recursively for however.

If this method succeeds in displaying the View, the flag indicating that the View needs to be displa self.

display, display::, drawSelf::, needsDisplay, update, displayFromOpaqueAncestor:::

displayFromOpaqueAncestor:(const NXRect *)rects
:(int)rectCount
:(BOOL)clipFlag

Correctly displays Views that aren't opaque. This method searches from the View up the View hie ancestor View. The rectangles specified by rects are copied and then converted to the opaque Vie display::: is sent to the opaque View. The third argument, clipFlag, is the same as the third argume

If the receiving View is opaque, this method has the same effect as display:::. Returns self.

display:::, isOpaque, setOpaque:

displayIfNeeded

Returns YES (the default) if the drawing that's generated by the View is clipped to the View's fra
NO.

setClipping:

```
dragFile:(const char *)filename
        fromRect:(NXRect *)rect
        slideBack:(BOOL) aFlag
        event:(NXEvent *)event
```

Allows a file icon to be dragged from the View to any application that accepts files. This method c
invoked from within an implementation of the mouseDown: method. The arguments are:

· filename is the complete name (including path) of the file to be dragged. If there is more than on
you must separate the filenames with a single tab (`\t') character.

·rect describes the position of the icon in the View's coordinates the width and height of rect must

·aFlag indicates whether the icon should slide back to its position in the View if the file is not acce
and filename is not accepted and the user has not disabled icon animation, the icon will slide ba
not.

·event is the mouse-down event record (or a copy).

This method returns self if the View successfully initiated the file dragging session otherwise it ret

dragImage:at:offset:event:pasteboard:source:slideBack:

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

Instigates an image-dragging session. This method only makes sense when invoked from within a
mouseDown: method. The arguments are

·anImage is the NXImage (contained within the View) that's being dragged.
·location is the NXImage's origin in the View's coordinate system.
·mouseOffset gives the mouse's current location relative to the mouse-down location.
·theMouseDown is the mouse-down that started the whole thing going (see below).
·pboard is the Pasteboard that holds the data that the NXImage represents (see below).
·sourceObject is the object that receives NXDraggingSource messages.
·slideFlag determines whether the NXImage should slide back if it's rejected.

Before invoking this method, the View must place the data that's being dragged on the drag pastel
must get the pasteboard, declare the type of data that it's placing, and then place the data:

Makes the View's coordinate system identical to that of its superview. This can reduce the amoun
that's generated to focus on the View. After invoking this method, the View's bounds rectangle or
frame rectangle origin.

Although the View's superview may be flipped, the View's coordinate system won't be flipped un
setFlipped: message. You should invoke drawInSuperview after creating the View and before app
transformations to it. Returns self.

setFlipped:

drawPageBorder:(float)width :(float)height

Allows applications that use the Application Kit pagination facility to draw additional marks on ea
method is invoked by beginPageSetupRect:placement:, and the default implementation doesn't dra
self.

drawSelf:(const NXRect *)rects :(int)rectCount

Implemented by subclasses to draw the View. Each View subclass must override this method to d
frame rectangle. The default implementation of this method does nothing.

This method is invoked by the display methods (display, display::, and display:::) you shouldn't se
message directly to a View.

rects is an array of rectangles indicating the region within the View that needs to be drawn. rectCo
number of rectangles in the rects array, which is either 1 or 3. If rectCount is 1, then rects[0] speci
drawn. If rectCount is 3, then rects[0] contains the smallest rectangle that completely encloses rec
two rectangles that actually specify the regions that need to be drawn. Note that if rectCount is 3,
contents of rects[0], or you can draw the contents of both rects[1] and rects[2], but there's no need
rectangles. For optimum drawing performance, you shouldn't draw anything that doesn't intersect
rectangles, although it is possible to draw the entire contents of the View and simply allow the con
clipped.

Your implementation of drawSelf:: doesn't need to invoke lockFocus focus is already locked on ar
to draw itself. Returns self.

display, display::, display:::

drawSheetBorder:(float)width :(float)height

Allows applications that use the Application Kit pagination facility to draw additional marks on ea
method is invoked by beginPageSetupRect:placement:, and the default implementation doesn't dra
self.

endHeaderComments

Writes out the end of a conforming PostScript header. It prints out the %%EndComments line and
prologue, including the Application Kit's standard printing package. The prologue should contain

beginPageSetupRect:placement:

endPageSetup

Writes the end of the page setup section, which begins with a %%EndPageSetup comment. This m
printPSCode: and faxPSCode: just after beginPageSetupRect:placement: is invoked. Returns self.

endPrologue

Writes out the end of the conforming PostScript prologue. This method is invoked by printPSCod
the prologue of the document has been written. Applications can override this method to add their
prologue. For example:

endPSOutput

Ends a print job. This method is invoked by printPSCode: and faxPSCode:. It closes the spool file
the old PostScript context so that further PostScript output is directed to the Window Server. Retu

beginPSOutput

endSetup

Writes out the end of the setup section, which begins with a %%EndSetup comment. This method
printPSCode: and faxPSCode: just after beginSetup is invoked. Returns self.

endTrailer

Writes the end of the conforming PostScript trailer. This method is invoked by printPSCode: and
beginTrailer is invoked. Returns self.

beginTrailer

faxPSCode:sender

Prints the View and all its subviews to a fax modem. If the user cancels the job, or if there are any
PostScript, this method returns nil otherwise it returns self.

This method normally brings up the Fax panel before actually initiating printing, but if sender impl

sendAt:(time_t)time
wantsCover:(BOOL)coverFlag
wantsNotify:(BOOL)notifyFlag
wantsHires:(BOOL)hiresFlag
faxName:(const char *)string

Sets up a fax session according to the arguments, and then faxes the View (and all its subviews).

findAncestorSharedWith:aView

Returns the closest common ancestor in the View hierarchy shared by aView and the receiving Vie
such ancestor. If aView and the receiving View are identical, this method returns self.

isDescendantOf:

findViewWithTag:(int)aTag

Returns the View's nearest descendant (including itself) that has the given tag, or nil if no matchin

tag

(float)frameAngle

Returns the angle of the View's frame relative to its superview's coordinate system.

rotateTo:, rotateBy:

free

Releases the storage for the View and all its subviews. This method also invalidates the cursor rec
window, frees the View's graphics state object (if any), and removes the View from the view hiera
longer be registered as a subview of any other View.

initFrame:

freeGState

Frees the graphics state object that was previously allocated for the View. Returns self.

allocateGState:

getBounds:(NXRect *)theRect

Copies the View's bounds rectangle into the structure specified by theRect. Returns self.

boundsAngle

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

Implemented by subclasses to determine the rectangle of the View to be printed for page number p
override this method to fill in theRect with the coordinates of the View (in its own coordinate syste
page requested. The View will later be told to display the theRect region in order to generate the i
This method is invoked by printPSCode: and faxPSCode: if the View's knowsPagesFirst:last: meth
View should not assume that the pages will be generated in any particular order.

This method returns YES if page is a valid page number for the View. It returns NO if page is outs

knowsPagesFirst:last:

(BOOL)getVisibleRect:(NXRect *)theRect

Gets the visible portion of the View. A rectangle enclosing the visible portion is placed in the stru
theRect. This method returns YES if part of the View is visible, and NO if none of it is.

Visibility is determined by intersecting the View's frame rectangle against the frame rectangles of
the view hierarchy, after appropriate coordinate transformations. Only those portions of the View
frame rectangles of all its ancestors can be visible.

If the View is in an off-screen window, or is covered by another window, this method may neverth
method does not take into account any siblings of the receiving View or siblings of its ancestors.

If the View is being printed, this method places the portion of the View that is visible on the page l
structure specified by theRect.

isVisible (Window), getDocVisibleRect: (ScrollView), getDocVisibleRect: (ClipView)

(int)gState

Returns the graphics state object allocated to the View. If no graphics state object has been allocat
not been focused on since receiving the allocateGState message, this method will return 0. Graphi
immediately allocated by invoking the allocateGState method, but are done in a ªlazyº fashion upo

allocateGState, lockFocus

(float)heightAdjustLimit

Returns the fraction (between 0.0 and 1.0) of the page that can be pushed onto the next page during
to prevent items from being cut in half. This limit applies to vertical pagination. This method is in
and faxPSCode:. By default, this method returns 0.2.

adjustPageHeightNew:top:bottom:limit:

hitTest:(NXPoint *)aPoint

Returns the subview of the receiving View that contains the point specified by aPoint. The lowest
hierarchy is returned. Returns the View if it contains the point but none of its subviews do, or nil i
within the receiving View.

Initializes the View, which must be a newly allocated View instance. This method does not alter th
rectangle, which is all zeros. This method is equivalent to initFrame:NULL. Note that if you insta
from Interface Builder, it will be initialized with the initFrame: method initialization code in the in
performed. Returns self.

initFrame:

initFrame:(const NXRect *)frameRect

Initializes the View, which must be a newly allocated View instance. The View's frame rectangle
that pointed to by frameRect. This method is the designated initializer for the View class, and can
View allocated from your own zone. Programs generally use instances of View subclasses rather t
the View class. Returns self.

init, + alloc (Object), + allocFromZone: (Object), + new (Object)

initGState

Implemented by subclasses of View to initialize the View's graphics state. The View will receive
previously sent it a notifyToInitGState:YES message. By default this method simply returns self,
to send PostScript code to initialize the View's graphics state. You could use this method to set a
width for the View. You should not use this method to send any coordinate transformations or clip

allocateGState, gState, notifyToInitGState:

invalidate:(const NXRect *)rects :(int)rectCount

Invalidates the View and its subviews for later display. This message is sent to the View after scro
subview of a ClipView and the View's parent ClipView previously received a setDisplayOnScroll
override this method to optimize drawing performance by accumulating the invalid areas for later
array of rectangles in the receiving View's coordinate system, and rectCount is the number of valid

If rectCount is 1, rects[0] specifies the region requiring redisplay. If rectCount is greater than 1, th
smallest rectangle that completely encloses the remaining rectangles in the rects array, which speci
requiring redisplay. Returns self.

rawScroll: (ClipView), display, display::, display:::, drawSelf::, setDisplayOnScroll: (ClipView

(BOOL)isAutodisplay

This method returns the View's automatic display status. After you change your data in such a wa
accurately represented, you should invoke this method to test the View's automatic display status.
enabled, you should send a display message to the View otherwise you should send it a setNeedsD

update, display, setAutodisplay, needsDisplay, setNeedsDisplay:, displayIfNeeded

(BOOL)isDescendantOf:aView

Views are not flipped.

setFlipped:

(BOOL)isFocusView

Returns YES if the receiving View is the View that's currently focused for drawing otherwise retu
words, returns YES if drawing commands will be drawn into this View.

lockFocus

(BOOL)isOpaque

Returns whether the View is opaque (as set through setOpaque:). Returns YES if the View guaran
completely cover the area within its frame when it draws itself otherwise returns NO.

setOpaque:, opaqueAncestor, displayFromOpaqueAncestor:::

(BOOL)isRotatedFromBase

Returns YES if the receiving View or any of its ancestors in the View hierarchy have been rotated

(BOOL)isRotatedOrScaledFromBase

Returns YES if the receiving View or any of its ancestors in the View hierarchy have been rotated
returns NO.

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

Indicates whether this View can return a rectangle specifying the region that must be displayed to p
method is invoked by printPSCode: and faxPSCode:. Just before invoking this method, the first pa
1, and the last page to be printed is set to the maximum integer size. You can therefore override th
first page to be printed, and also the last page to be printed if the View knows where its pages lie.
YES, the printing mechanism will later query the View for the rectangle corresponding to a specifi
forPage:.

getRect:forPage:

(BOOL)lockFocus

Locks the PostScript focus on the View so that subsequent graphics commands are applied to the V
ensures that the View draws in the correct coordinates and to the correct device. You must send th
before you draw to it, and you must balance it with an unlockFocus message to the View when you
Returns YES if the focus was already locked on the View, and NO if it wasn't.

and aRect must be expressed in the same coordinate system.

You should never use the NXPointInRect() function as a substitute for this method.

convertPoint:fromView:, NXMouseInRect(), NXPointInRect()

moveBy:(NXCoord)deltaX :(NXCoord)deltaY

Moves the origin of the View's frame rectangle by (deltaX, deltaY) in its superview's coordinates. through the moveTo:: method. Returns self.

moveTo::, sizeBy::

moveTo:(NXCoord)x :(NXCoord)y

Moves the origin of the View's frame rectangle to (x, y) in its superview's coordinates. This meth descendantFrameChanged: message to the View's superview. Returns self.

setFrame:, sizeTo::, descendantFrameChanged:

(BOOL)needsDisplay

Returns whether the View needs to be displayed to reflect changes to its contents. If automatic dis View will not redisplay itself automatically, so you can invoke this method to determine whether y display message to the View. The flag indicating that the View needs to be displayed is cleared by when the View is displayed.

setNeedsDisplay:, update, setAutodisplay, isAutodisplay, display, displayIfNeeded

notifyAncestorWhenFrameChanged:(BOOL)flag

Determines whether the receiving View will inform its ancestors in the view hierarchy whenever it flag is YES, subsequent sizeTo:: and moveTo:: messages to the View will send a descendantFrame the view hierarchy. If flag is NO, no descendantFrameChanged: message will be sent to the View' descendantFrameChanged: message permits Views to make any necessary adjustments when a sub moved. Returns self.

descendantFrameChanged:, sizeTo::, moveTo::

notifyToInitGState:(BOOL)flag

Determines whether the View will be sent initGState messages to allow it to initialize new graphic is YES, initGState messages will be sent to the View at the appropriate time otherwise, they will n View is not sent messages to initialize its graphics state objects. Returns self.

initGState

opaqueAncestor

Returns the View's closest opaque ancestor (including the receiving View itself).

isOpaque, displayFromOpaqueAncestor:::

openSpoolFile:(char *)filename

Opens the filename file for print spooling. This method is invoked by printPSCode: and faxPSCod directly invoked in program code. However, you can override it to modify its behavior.

If filename is NULL or an empty string, the PostScript code is sent directly to the printing daemon a file. (However, if the Window is being previewed or saved, a default file is opened in /tmp).

If filename is provided, the file is opened. The printing machinery will then write the PostScript c file will be printed (or faxed) using lpr.

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

(BOOL)performKeyEquivalent:(NXEvent *)theEvent

Implemented by subclasses of View to allow them to respond to keyboard input. If the View respo should take the appropriate action and return YES. Otherwise, it should return the result of passing super, which will pass the message down the View hierarchy:

commandKey: (Window and Panel)

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

Determines the location of the rectangle being printed on the physical page. This method is invoke faxPSCode:. aRect is the rectangle being printed on the current page. This method sets location to rectangle from the lower left corner of the page. All coordinates are in the default PostScript coord page.

By default, if the flags for centering are YES in the global PrintInfo object, this routine centers the margins. If the flags are NO, it defaults to abutting the rectangle against the top left margin. Retu

printPSCode:sender

Prints the View and all its subviews. If the user cancels the job, or if there are any errors in genera code, this method returns nil otherwise it returns self.

Reads the View and its subviews from the typed stream stream. Returns self.

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

Registers the Pasteboard types that the View will accept in an image-dragging session. pbTypes i
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 View as accepting files:

unregisterDraggedTypes

removeCursorRect:(const NXRect *)aRect cursor:anNXCursor

Removes a cursor rectangle from the View. aRect and anNXCursor must match the values that we
cursor rectangle was added (through addCursorRect:cursor:).

You rarely need to use this method it's usually easier to use Window's invalidateCursorRectsForV
resetCursorRects mechanism restore the cursor rectangles. Returns self.

invalidateCursorRectsForView: (Window), resetCursorRects

removeFromSuperview

Unlinks the View from its superview and its Window, removes it from the responder chain, and in
rectangles. Returns self.

addSubview:

renewGState

Forces the View to reinitialize its graphics state object. This method is lazy the graphics state obje
the View actually draws. Returns self.

replaceSubview:oldView with:newView

Replace oldView with newView in the View's subview list. This method does nothing and returns
subview of the View or if newView is not a View. Otherwise, this method returns oldView.

addSubview:

addCursorRect:, invalidateCursorRectsForView: (Window)

resizeSubviews:(const NXSize *)oldSize

Informs the View's subviews that the View's bounds rectangle size has changed. This method is i
sizeTo:: method if the View has subviews and has received a setAutoresizeSubviews:YES messag
method sends a superviewSizeChanged: message to each subview. You should not invoke this me
may want to override it to define a specific retiling behavior. oldSize is the previous bounds recta

sizeTo::, setAutoresizeSubviews:, superviewSizeChanged:

rotate:(NXCoord)angle

Rotates the View's drawing coordinates by angle degrees from its current angle of orientation. Po
counterclockwise rotation negative values indicate clockwise rotation. The position of the coordin
remains unchanged it's at the center of the rotation. Returns self.

translate::, scale::, setDrawRotation:

rotateBy:(NXCoord)deltaAngle

Rotates the View's frame rectangle by deltaAngle degrees from its current angle of orientation. Po
frame in a counterclockwise direction negative values rotate it clockwise. The position of the fram
remains unchanged it's at the center of the rotation. Returns self.

rotateTo:

rotateTo:(NXCoord)angle

Rotates the View's frame rectangle to angle degrees in its superview's coordinate system. The pos
rectangle origin remains unchanged it's at the center of the rotation. Returns self.

rotateBy:

scrollPoint:(const NXPoint *)aPoint

Scrolls the View, which must be a ClipView's document view. aPoint is given in the receiving Vi
the scroll, aPoint will be coincident with the bounds rectangle origin of the ClipView, which is its
upper left corner if the receiving View is flipped. Returns self.

setDocView: (ClipView)

scrollRect:(const NXRect *)aRect by:(const NXPoint *)delta

Scrolls the aRect rectangle, which is expressed in the View's drawing coordinates, by delta. Only
visible before and after scrolling are moved. This method works for all Views and does not requir
immediate ancestor be a ClipView or ScrollView. Returns self.

scrollRectToVisible:(const NXRect *)aRect

Scrolls aRect so that it becomes visible within the View's parent ClipView. The receiving View m
document view. This method will scroll the ClipView the minimum amount necessary to make aR
rectangle in the receiving View's coordinates. Returns self if scrolling actually occurs otherwise r

setDocView: (ClipView)

setAutodisplay:(BOOL)flag

Enables or disables automatic display of the View. If flag is YES, subsequent messages to the Vie
appearance are automatically reflected on the screen. If flag is NO, you must explicitly send a disp
changes to the View. By default, changes are automatically displayed. If automatic display is disa
a dirty flag which you can query with the needsDisplay method to determine whether you need to
message. Returns self.

isAutodisplay, needsDisplay, setNeedsDisplay:, display, update, displayIfNeeded

setAutoresizeSubviews:(BOOL)flag

Determines whether the resizeSubviews: message will be sent to the View upon receipt of a sizeTo
default, automatic resizing of subviews is disabled. Returns self.

resizeSubviews:, sizeTo::, superviewSizeChanged:

setAutosizing:(unsigned int)mask

Determines how the receiving View's frame rectangle will change when its superview's size chang
ORing the following together:

setClipping:(BOOL)flag

Determines whether drawing is clipped to the View's frame rectangle. Views are clipped by defau
View won't draw outside its frame, you can turn off clipping to reduce the amount of PostScript co
Server. You can also use this method to enable clipping in a View that inherits from a subclass tha
You should send a setClipping: message to the View before it first draws, usually from the method
View. Returns self.

lockFocus, drawInSuperview, initFrame:, doesClip

setDrawOrigin:(NXCoord)x :(NXCoord)y

Shifts the View's coordinate system so that (x, y) corresponds to the same point as the View's fram
the View's coordinates have been rotated or flipped, this won't necessarily coincide with its bound
Returns self.

translate::, setDrawSize::, setDrawRotation:

setDrawRotation:(NXCoord)angle

Rotates the View's coordinate system around its frame rectangle origin so that angle defines the re
View's frame rectangle and its drawing coordinates. Returns self.

rotate:, setDrawOrigin::, setDrawSize::

setDrawSize:(NXCoord)width :(NXCoord)height

Scales the View's coordinate system so that width and height define the size of the View's frame r
coordinates. If the View's drawing coordinates have been rotated, the View's frame rectangle size
the same as its bounds rectangle size. Returns self.

scale::, setDrawOrigin::, setDrawRotation:

setFlipped:(BOOL)flag

Sets the direction of the View's y-axis. If flag is YES, the View's origin will be its upper left corn
values will increase towards the bottom of the View. If flag is NO, the origin is the bottom left co
to the top this is the default configuration for a View.

Although a View is positioned in its superview's coordinate system, no View will have a flipped c
it receives a setFlipped:YES message of its own it doesn't inherit flipped coordinates from its supe

This method may also send a descendantFlipped: message to the receiving View's superview. Ret

notifyWhenFlipped:, descendantFlipped:, initFrame:, isFlipped

setNeedsDisplay:(BOOL)flag

This method sets a flag indicating whether the View needs to be displayed. After the View change such a way that it's no longer accurately reflected on the screen, it should query itself with an isAu automatic display is enabled, the View should send a display message to itself. If automatic displa should send a setNeedsDisplay:YES message to itself. This message has no effect if automatic dis Returns self.

update, setAutodisplay, isAutodisplay, needsDisplay:, display:::, displayIfNeeded

setOpaque:(BOOL)flag

Registers whether the View is opaque. If a View can guarantee that it will completely cover the ar (with opaque paint), it should send itself a setOpaque:YES message (typically, as part of its initiali make drawing in the view hierarchy more efficient. Returns self.

isOpaque, opaqueAncestor, displayFromOpaqueAncestor:::

(BOOL)shouldDelayWindowOrderingForEvent:(NXEvent *)anEvent

Returns YES if the normal Window ordering and activation mechanism should be delayed until the You never inovke this method directly it's invoked automatically for each mouse-down that's direc default implementation returns NO.

A View subclass that contains draggable images should implement this to return YES (perhaps pre on the data in anEvent, the event record for the mouse-down itself). This allows the user to click o without bringing the View's Window to the front or making its application active. Note that this m this ordering and activation from occurring, it simply puts it off until the user releases the mouse. and activation to be skipped when the mouse is released, the View should send a preventWindowO Application object from within its implementation of mouseDown:. The preventWindowOrdering automatically by View's dragImage:... methodÐin other words, ordering and activation is prevent drags the clicked-on item.

(BOOL)shouldDrawColor

Returns whether the View should be drawn using color. If the View is being drawn to a window a store color, this method returns NO otherwise it returns YES.

sizeBy:(NXCoord)deltaWidth :(NXCoord)deltaHeight

Resizes the View by deltaWidth and deltaHeight in its superview's coordinates. This method work sizeTo:: method. Returns self.

sizeTo::, moveBy::

sizeTo:(NXCoord)width :(NXCoord)height

self.

subviews

Returns the List object that contains the receiving View's subviews. You can use this List to send in the View hierarchy. You never modify this List directly use addSubview: and removeFromSup remove Views from the View hierarchy. If the View has no subviews (and never did), nil is return that have all since been removed, an empty List is returned.

superview, addSubview:, removeFromSuperview

superview

Returns the View's superview. If the View hasn't a superview, nil is returned. When applying thi you should check the return value against the content View of the View's Window to avoid flying hierarchy.

window, subviews, addSubview:, removeFromSuperview

superviewSizeChanged:(const NXSize *)oldSize

Informs the View that its superview's size has changed. This method is invoked when the View's resizeSubviews: message. This method will automatically resize the View according to the param setAutosizing: message. You may want to override this method to provide specific resizing behav previous bounds rectangle size of the receiving View's superview. Returns self.

resizeSubviews:, sizeTo::, setAutoresizeSubviews:

suspendNotifyAncestorWhenFrameChanged:(BOOL)flag

Temporarily disables or reenables the sending of descendantFrameChanged: messages to the View View is sized or moved. You must have previously sent the View a notifyAncestorWhenFrameCh for this method to have any effect. These messages do not nest. Returns self.

descendantFrameChanged:, notifyAncestorWhenFrameChanged:, sizeTo::, moveTo::,

(int)tag

Returns the View's tag, a integer that you can use to identify objects in your application. By defau You can override this method to identify certain Views. For example, your application could take View with a given tag receives a mouse event.

findViewWithTag:

translate:(NXCoord)x :(NXCoord)y

lockFocus, display:::

unregisterDraggedTypes

Unregisters the View as a possible recipient of dragged-images.

registerForDraggedTypes:count:

update

Invokes the proper update behavior when the contents of the View have been changed in such a wa
longer accurately represented on the screen. If automatic display is enabled, this method invokes d
method sets a flag indicating that the View needs to be displayed. Returns self.

setNeedsDisplay,