

NSView

Inherits From: NSResponder : NSObject
Declared In: appkit/NSView.h, appkit/NSClipView.h (NSClipViewSuperview)

Initializing NSView Objects

- (id)**initWithFrame:**(CGRect)*frameRect* Initializes a new NSView object to the location and dimensions of *frameRect*.

Managing the NSView Hierarchy

- (void)**addSubview:**(NSView *)*aView* Makes *aView* a subview of the receiving view object.
- (void)**addSubview:**(NSView *)*aView*
 positioned:(NSWindowOrderingMode)*place* Makes *aView* a subview of the receiving view object.
 relativeTo:(NSView *)*otherView* It is positioned relative to *otherView* according to *place*.
- (NSView *)**ancestorSharedWithView:**(NSView *)*aView* Returns the ancestor view shared by *aView* and the receiver; **self** if *aView* is the receiving view or if the receiving view is the ancestor of *aView*; *aView* if it is the superview of the receiving view; or **nil** in any other case.
- (BOOL)**isDescendantOf:**(NSView *)*aView* Returns whether *aView* is an ancestor of the receiver.
- (NSView *)**opaqueAncestor** Returns the receiver's nearest opaque ancestor.
- (void)**removeFromSuperview** Removes the receiver from the view hierarchy.
- (void)**replaceSubview:**(NSView *)*oldView*
 with:(NSView *)*newView* Replaces *oldView* with *newView*.
- (void)**sortSubviewsUsingFunction:**(int (*)(id ,id ,void *))*compare*
 context:(void *)*context* Sorts the receiving view's subviews using the sorting function *compare* and

- (NSArray *)**subviews**
- (NSView *)**superview**
- (NSWindow *)**window**
- (void)**viewWillMoveToWindow:**(NSWindow *)*newWindow*

the context *context*. The first two arguments of the function are the views to be compared.

Returns a mutable array of the receiving view object's subviews.

Returns the receiving view object's superview.

Returns the window in which the view is displayed.

Notifies the view that it will move to a new window.

Modifying the Frame Rectangle

- (float)**frameRotation**
- (CGRect)**frame**
- (void)**rotateByAngle:**(float)*angle*
- (void)**setFrame:**(CGRect)*frameRect*
- (void)**setFrameOrigin:**(CGPoint)*newOrigin*
- (void)**setFrameRotation:**(float)*angle*
- (void)**setFrameSize:**(CGSize)*newSize*

Returns the angle of the frame rectangle's rotation.

Gets the view's frame rectangle.

Rotates the view's frame rectangle by *angle*.

Assigns the view a new frame rectangle.

Sets the origin of the view's frame to *newOrigin*.

Rotates the view's frame to *angle*.

Resizes the view's frame to *newSize*.

Modifying the Coordinate System

- (float)**boundsRotation**
- (CGRect)**bounds**
- (BOOL)**isFlipped**
- (BOOL)**isRotatedFromBase**
- (BOOL)**isRotatedOrScaledFromBase**
- (void)**scaleUnitSquareToSize:**(CGSize)*newSize*
- (void)**setBounds:**(CGRect)*aRect*
- (void)**setBoundsOrigin:**(CGPoint)*newOrigin*
- (void)**setBoundsRotation:**(float)*angle*
- (void)**setBoundsSize:**(CGSize)*newSize*
- (void)**translateOriginToPoint:**(CGPoint)*point*

Returns the rotation of the view's coordinate system.

Gets the view's bounds rectangle.

Returns whether the view is flipped.

Returns whether the view is rotated.

Returns whether the view is rotated or scaled.

Scales the NSView's coordinate system unit size to *newSize*.

Sets the NSView's bounds rectangle to *aRect*.

Sets the NSView's drawing origin to *newOrigin*.

Rotates the NSView's coordinate system to *angle*.

Resizes the NSView's coordinate system to *newSize*.

Shifts the NSView's coordinate system to *point*.

Converting Coordinates

- (CGRect)**centerScanRect:**(CGRect)*aRect*

Converts the rectangle *aRect* to lie on centers of pixels.

- (NSPoint)**convertPoint:(NSPoint)aPoint
fromView:(NSView *)aView** Converts *aPoint* in *aView* to the receiver's coordinates.
- (NSPoint)**convertPoint:(NSPoint)aPoint
toView:(NSView *)aView** Converts *aPoint* in the receiver to *aView*'s coordinates.
- (NSRect)**convertRect:(NSRect)aRect
fromView:(NSView *)aView** Converts the rectangle *aRect* in *aView* to the receiver's coordinates.
- (NSRect)**convertRect:(NSRect)aRect
toView:(NSView *)aView** Converts the rectangle *aRect* in the receiver to *aView*'s coordinates.
- (NSSize)**convertSize:(NSSize)aSize
fromView:(NSView *)aView** Converts *aSize* in *aView* to the receiver's coordinates.
- (NSSize)**convertSize:(NSSize)aSize
toView:(NSView *)aView** Converts *aSize* in the receiver to *aView*'s coordinates.

Notifying Ancestor Views

- (BOOL)**postsFrameChangedNotifications** Returns whether notification of frame changes to ancestors is activated.
- (void)**setPostsFrameChangedNotification:(BOOL)flag** Sets whether to activate ancestor notification.

Resizing Subviews

- (void)**resizeSubviewsWithOldSize:(NSSize)oldSize** Initiates **superviewSizeChanged:** messages to subviews.
- (void)**setAutoresizesSubviews:(BOOL)flag** Sets whether to notify subviews of resizing.
- (BOOL)**autoresizesSubviews** Returns whether the NSView notifies subviews of resizing.
- (void)**setAutoresizingMask:(unsigned int)mask** Determines automatic resizing behavior.
- (unsigned int)**autoresizingMask** Returns the NSView's autosizing mask.
- (void)**resizeWithOldSuperviewSize:(NSSize)oldSize** Notifies subviews that the superview changed size.

Graphics State Objects

- (void)**allocateGState** Allocates a graphics state object.
- (void)**releaseGState** Release the NSView's graphics state object.
- (int)**gState** Returns the NSView's graphics state object.

- (void)**renewGState** Marks the NSView's graphics state object as needing initialization.
- (void)**setUpGState** Sets up the NSView's graphics state object.

Focusing

- + (NSView *)**focusView** Returns the currently focused view.
- (void)**lockFocus** Brings the receiving view into focus.
- (void)**unlockFocus** Unfocuses the receiving view.

Displaying

- (BOOL)**canDraw** Returns whether the view object can draw.
- (void)**display** Displays the receiving view and its subviews.
- (void)**displayIfNeeded** Conditionally displays the receiving view and its subviews (if opaque).
- (void)**displayIfNeededIgnoringOpacity** Conditionally displays the receiving view and its subviews, regardless of opacity.
- (void)**displayRect:(NSRect)aRect** Displays the receiving view and its subviews (if opaque) within *aRect*.
- (void)**displayRectIgnoringOpacity:(NSRect)aRect** Displays the receiving view and its subviews (regardless of opacity) within *aRect*.
- (void)**drawRect:(NSRect)rect** Implemented by subclasses to supply drawing instructions.
- (NSRect)**visibleRect** Gets the receiving view's visible portion.
- (BOOL)**isOpaque** Returns whether the view is opaque.
- (BOOL)**needsDisplay** Returns whether the view needs to be redisplayed.
- (void)**setNeedsDisplay:(BOOL)flag** If *flag* is YES, marks the view as changed, needing redisplay.
- (void)**setNeedsDisplayInRect:(NSRect)invalidRect** Marks the view as changed, needing redisplay in rectangle *invalidRect*.
- (BOOL)**shouldDrawColor** Returns whether the view should be drawn in color.

Scrolling

- (NSRect)**adjustScroll:(NSRect)newVisible** Lets the view object adjust the visible rectangle.
- (BOOL)**autoscroll:(NSEvent *)theEvent** Scrolls in response to a mouse-dragged event.
- (void)**reflectScrolledClipView:(NSClipView *)aClipView** Reflects scrolling within clip view *aClipView*.
- (void)**scrollClipView:(NSClipView *)aClipView toPoint:(NSPoint)aPoint** Scrolls the clip view *aClipView* to *aPoint*.

- (void)**scrollPoint:**(NSPoint)*aPoint* Aligns *aPoint* with the content view's origin.
- (void)**scrollRect:**(NSRect)*aRect*
by:(NSSize)*delta* Shifts the rectangle *aRect* by *delta*.
- (BOOL)**scrollRectToVisible:**(NSRect)*aRect* Scrolls the view so the rectangle *aRect* is visible.

Managing the Cursor

- (void)**addCursorRect:**(NSRect)*aRect*
cursor:(NSCursor *)*anObject* Adds a cursor rectangle *aRect* for cursor *anObject* to the NSView.
- (void)**discardCursorRects** Removes all cursor rectangles in the view.
- (void)**removeCursorRect:**(NSRect)*aRect*
cursor:(NSCursor *)*anObject* Removes cursor rectangle *aRect* for cursor *anObject* from the view.
- (void)**resetCursorRects** Implemented by subclasses to reset their cursor rectangles.

Assigning a Tag

- (int)**tag** Returns the view object's tag.
- (id)**viewWithTag:**(int)*aTag* Returns the subview object with *aTag* as its tag.

Aiding Event Handling

- (BOOL)**acceptsFirstMouse:**(NSEvent *)*theEvent* Returns whether the view object accepts first mouse-down events.
- (NSView *)**hitTest:**(NSPoint)*aPoint* Returns the lowest subview containing the point *aPoint*.
- (BOOL)**mouse:**(NSPoint)*aPoint*
inRect:(NSRect)*aRect* Returns whether the point *aPoint* lies inside the *aRect*.
- (BOOL)**performKeyEquivalent:**(NSEvent *)*theEvent* Implemented by subclasses to perform key-equivalent commands. Returns whether a subview handled *theEvent*.
- (void)**removeTrackingRect:**(NSTrackingRectTag)*tag* Removes the tracking rectangle identified by *tag* from the view. (*tag* is a unique identifier returned from the **addTractingRect:owner:assumeInside:** method.)
- (BOOL)**shouldDelayWindowOrderingForEvent:**(NSEvent *)*anEvent*

Returns whether the view's window is brought forward normally (mouse-down) or delayed (mouse-up).

- (NSTrackingRectTag)**addTrackingRect:**(NSRect)*aRect*

owner:(id)*anObject*

assumeInside:(BOOL)*flag*

Adds a tracking rectangle (*aRect*) owned by *anObject* to the receiving NSView. *flag* indicates whether the tracking rectangle will be only inside the NSView. Returns a unique tag that identifies the tracking rectangle.

Dragging

- (BOOL)**dragFile:**(NSString *)*filename*

fromRect:(NSRect)*rect*

slideBack:(BOOL)*slideFlag*

event:(NSEvent *)*event*

Initiates a file-dragging session, dragging file indicated by path *filename*. *rect* describes the position of the icon in the View's coordinates. *slideFlag* determines whether the NXImage should slide back if rejected

- (void)**dragImage:**(NSImage *)*anImage*

at:(NSPoint)*viewLocation*

offset:(NSSize)*initialOffset*

event:(NSEvent *)*event*

pasteboard:(NSPasteboard *)*pboard*

source:(id)*sourceObject*

slideBack:(BOOL)*slideFlag*

Initiates an image-dragging session, dragging *anImage* from *viewLocation*. *initialOffset* is the difference in the mouse location from the mouse-down. *pboard* is the pasteboard holding the data. *sourceObject* is the object receiving NXDraggingSource messages. *slideFlag* determines whether the NXImage should slide back if rejected.

- (void)**registerForDraggedTypes:**(NSArray *)*newTypes*

Registers the pasteboard types that the window will accept in an image-dragging session.

- (void)**unregisterDraggedTypes**

Unregisters the window as a recipient of dragged images.

Printing

- (NSData *)**dataWithEPSInsideRect:**(NSRect)*aRect*

Returns a data object initialized with the EPS data within *aRect* in the receiving view.

- (void)**print:**(id)*sender*

Prints the view and its subviews.

- (void)**writeEPSInsideRect:**(NSRect)*rect*

toPasteboard:(NSPasteboard *)*pasteboard*

Places PostScript code for the rectangle *rect* on the *pasteboard*.

Pagination

- (void)**adjustPageHeightNew:**(float *)*newBottom* Assists automatic pagination of the view object.
 top:(float)*oldTop*
 bottom:(float)*oldBottom*
 limit:(float)*bottomLimit*
- (void)**adjustPageWidthNew:**(float *)*newRight* Assists automatic pagination of the view object.
 left:(float)*oldLeft*
 right:(float)*oldRight*
 limit:(float)*rightLimit*
- (float)**heightAdjustLimit** Returns how much of a page can go on the next page.
- (BOOL)**knowsPagesFirst:**(int *)*firstPageNum* Returns whether the view paginates itself.
 last:(int *)*lastPageNum*
- (NSPoint)**locationOfPrintRect:**(NSRect)*aRect* Locates the printing rectangle on the page.
- (NSRect)**rectForPage:**(int)*page* Provides how much of the view will print on page.
- (float)**widthAdjustLimit** Returns how much of a page can go on the next page.

Writing Conforming PostScript

- (void)**addToPageSetup** Allows you to adjust for differences in the graphics state between the screen and the printer.
- (void)**beginPage:**(int)*ordinalNum* Writes a page separator.
 label:(NSString *)*aString*
 bBox:(NSRect)*pageRect*
 fonts:(NSString *)*fontNames*
- (void)**beginPageSetupRect:**(NSRect)*aRect* Writes the beginning of a page setup section.
 placement:(NSPoint)*location*
- (void)**beginPrologueBBox:**(NSRect)*boundingBox* Writes the header for a print job.
 creationDate:(NSString *)*dateCreated*
 createdBy:(NSString *)*anApplication*
 fonts:(NSString *)*fontNames*
 forWhom:(NSString *)*user*
 pages:(int)*numPages*
 title:(NSString *)*aTitle*

- (void)**beginSetup** Writes the beginning of the job setup section.
- (void)**beginTrailer** Writes the beginning of the trailer for the print job.
- (void)**drawPageBorderWithSize:(NSSize)*borderSize***
Implemented by subclasses to draw in margins (e.g., borders, numbering).
borderSize is the size of the border.
- (void)**drawSheetBorderWithSize:(NSSize)*borderSize***
Implemented by subclasses to draw in margins (e.g., borders, numbering).
borderSize is the size of the border.
- (void)**endHeaderComments** Writes the end of the header.
- (void)**endPrologue** Writes the end of the prologue.
- (void)**endSetup** Writes the end of the job setup section.
- (void)**endPageSetup** Writes the end of a page setup section.
- (void)**endPage** Writes the end of a page.
- (void)**endTrailer** Writes the end of the trailer.