

<code>initWithFrame:(CGRect)frameRect</code>	Initializes a new <code>NSView</code> object to the location and dimensions of <code>frameRect</code> .
<code>-(void)addSubview:(NSView *)aView</code>	Makes <code>aView</code> a subview of the receiving view object.
<code>-(void)addSubview:(NSView *)aView positioned:(NSWindowOrderingMode)place relativeTo:(NSView *)otherView</code>	Makes <code>aView</code> a subview of the receiving view object. It is positioned relative to <code>otherView</code> according to place.
<code>-(NSView *)ancestorSharedWithView:(NSView *)aView</code>	Returns the ancestor view shared by <code>aView</code> and the receiver self if <code>aView</code> is the receiving view or if the receiving view is the ancestor of <code>aView</code> <code>aView</code> if it is the superview of the receiving view or nil in any other case.
<code>-(BOOL)isDescendantOf:(NSView *)aView</code>	Returns whether <code>aView</code> is an ancestor of the receiver.
<code>-(NSView *)opaqueAncestor</code>	Returns the receiver's nearest opaque ancestor.
<code>-(void)removeFromSuperview</code>	Removes the receiver from the view hierarchy.
<code>-(void)replaceSubview:(NSView *)oldView with:(NSView *)newView</code>	Replaces <code>oldView</code> with <code>newView</code> .
<code>-(void)sortSubviewsUsingFunction:(int (*)(id ,id ,void *))compare context:(void *)context</code>	Sorts the receiving view's subviews using the sorting function <code>compare</code> and the context <code>context</code> . The first two arguments of the function are the views to be compared.
<code>-(NSArray *)subviews</code>	Returns a mutable array of the receiving view object's subviews.
<code>-(NSView *)superview</code>	Returns the receiving view object's superview.
<code>-(NSWindow *)window</code>	Returns the window in which the view is displayed.
<code>-(void)viewWillMoveToWindow:(NSWindow *)newWindow</code>	Notifies the view that it will move to a new window.
<code>-(float)frameRotation</code>	Returns the angle of the frame rectangle's rotation.
<code>-(CGRect)frame</code>	Gets the view's frame rectangle.
<code>-(void)rotateByAngle:(float)angle</code>	Rotates the view's frame rectangle by <code>angle</code> . This method posts the <code>NSViewFocusChangedNotification</code> notification with the receiving object to the default notification center.

	NSNotification and NSViewFocusChangedNotification with the receiving object to the default notification center.
(CGFloat)boundsRotation	Returns the rotation of the view's coordinate system.
(CGRect)bounds	Gets the view's bounds rectangle.
(BOOL)isFlipped	Returns whether the view is flipped.
(BOOL)isRotatedFromBase	Returns whether the view is rotated.
(BOOL)isRotatedOrScaledFromBase	Returns whether the view is rotated or scaled.
(void)scaleUnitSquareToSize:(NSSize)newSize	Scales the NSView's coordinate system unit size to newSize. This method sends the notification NSViewFocusChangedNotification with the receiving object to the default notification center.
(void)setBounds:(CGRect)aRect	Sets the NSView's bounds rectangle to aRect.
(void)setBoundsOrigin:(NSPoint)newOrigin	Sets the NSView's drawing origin to newOrigin. This method sends the NSViewFocusChangedNotification notification with the receiving object to the default notification center.
(void)setBoundsRotation:(CGFloat)angle	Rotates the NSView's coordinate system to angle. This method sends the NSViewFocusChangedNotification notification with the receiving object to the default notification center.
(void)setBoundsSize:(NSSize)newSize	Resizes the NSView's coordinate system to newSize. This method sends the NSViewFocusChangedNotification notification with the receiving object to the default notification center.
(void)translateOriginToPoint:(NSPoint)point	Shifts the NSView's coordinate system to point. This method sends the NSViewFocusChangedNotification notification with the receiving object to the default notification center.
(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.
(CGRect)convertRect:(CGRect)aRect fromView:(NSView *)aView	Converts the rectangle aRect in aView to the receiver's coordinates.
(CGRect)convertRect:(CGRect)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.
(BOOL)postsFrameChangedNotifications	Returns whether notifications of frame changes to ancestors are posted.

<code>(BOOL)autoresizesSubviews</code>	Returns whether the <code>NSView</code> notifies subviews of resizing.
<code>(void)setAutoresizingMask:(unsigned int)mask</code>	Determines automatic resizing behavior.
<code>(unsigned int)autoresizingMask</code>	Returns the <code>NSView</code> 's autosizing mask.
<code>(void)resizeWithOldSuperviewSize:(NSSize)oldSize</code>	Notifies subviews that the superview changed size.
<code>(void)allocateGState</code>	Allocates a graphics state object.
<code>(void)releaseGState</code>	Release the <code>NSView</code> 's graphics state object.
<code>(GState)gState</code>	Returns the <code>NSView</code> 's graphics state object.
<code>(void)renewGState</code>	Marks the <code>NSView</code> 's graphics state object as needing initialization.
<code>(void)setUpGState</code>	Sets up the <code>NSView</code> 's graphics state object.
<code>(void)lockFocus</code>	Brings the receiving view into focus.
<code>(void)unlockFocus</code>	Unfocuses the receiving view.
<code>(BOOL)canDraw</code>	Returns whether the view object can draw.
<code>(void)display</code>	Displays the receiving view and its subviews.
<code>(void)displayIfNeeded</code>	Conditionally displays the receiving view and its subviews.
<code>(void)displayIfNeededIgnoringOpacity</code>	Conditionally displays the receiving view and its subviews (regardless of opacity).
<code>(void)displayRect:(NSRect)aRect</code>	Displays the receiving view and its subviews (if opaque) within the rectangle <code>aRect</code> .
<code>(void)displayRectIgnoringOpacity:(NSRect)aRect</code>	Displays the receiving view and its subviews (regardless of opacity) within the rectangle <code>aRect</code> .
<code>(void)drawRect:(NSRect)rect</code>	Implemented by subclasses to supply drawing instructions for the rectangle <code>rect</code> .
<code>(NSRect)visibleRect</code>	Gets the receiving view's visible portion.
<code>(BOOL)isOpaque</code>	Returns whether the view is opaque.
<code>(BOOL)needsDisplay</code>	Returns whether the view needs to be redisplayed.
<code>(void)setNeedsDisplay:(BOOL)flag</code>	If flag is YES, marks the view as changed, needing redisplay.
<code>(void)setNeedsDisplayInRect:(NSRect)invalidRect</code>	Marks the view as changed, needing redisplay in rectangle <code>invalidRect</code> .
<code>(BOOL)shouldDrawColor</code>	Returns whether the view should be drawn in color.
<code>(NSRect)adjustScroll:(NSRect)newVisibleRect</code>	Lets the view object adjust the visible rectangle.
<code>(BOOL)autoscroll:(NSEvent *)theEvent</code>	Scrolls in response to a mouse-dragged event.

`(BOOL)scrollRectToVisible:(NSRect)aRect`

Scrolls the view so the rectangle aRect is visible.

`(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

`(int)tag`

Returns the view object's tag.

`(NSView *)viewWithTag:(int)aTag`

Returns the subview object with aTag as its tag.

`(BOOL)acceptsFirstMouse:(NSEvent *)theEvent`

Returns whether the view object accepts first mouse-down

`(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 com  
whether a subview handled theEvent.

`(void)removeTrackingRect:(NSTrackingRectTag)tag`

Removes the tracking rectangle identified by tag from the  
identifier returned from the addTractingRect:owner:ass

`(BOOL)shouldDelayWindowOrderingForEvent:(NSEvent *)anEvent`

Returns whether the view's window is brought forward no  
or delayed (mouse-up).

`(NSTrackingRectTag)addTrackingRect:(NSRect)aRect  
owner:(id)anObject  
userData:(void *)data  
assumeInside:(BOOL)flag`

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

`(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 NSImage should slide back if rejected

`(void)dragImage:(NSImage *)anImage  
at:(NSPoint)viewLocation  
offset:(NSSize)initialOffset  
event:(NSEvent *)event  
pasteboard:(NSPasteboard *)pboard`

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

<code>NSData *)dataWithEPSInsideRect:(NSRect)aRect</code>	Returns a data object initialized with the EPS data within a view.
<code>(id)fax:(id)sender</code>	Faxes the view and its subviews.
<code>(id)print:(id)sender</code>	Prints the view and its subviews.
<code>(id)writeEPSInsideRect:(NSRect)rect toPasteboard:(NSPasteboard *)pasteboard</code>	Places PostScript code for the rectangle rect on the pasteboard.
<code>(id)adjustPageHeightNew:(float *)newBottom top:(float)oldTop bottom:(float)oldBottom limit:(float)bottomLimit</code>	Assists automatic pagination of the view object.
<code>(id)adjustPageWidthNew:(float *)newRight left:(float)oldLeft right:(float)oldRight limit:(float)rightLimit</code>	Assists automatic pagination of the view object.
<code>(float)heightAdjustLimit</code>	Returns how much of a page can go on the next page.
<code>(BOOL)knowsPagesFirst:(int *)firstPageNum last:(int *)lastPageNum</code>	Returns whether the view paginates itself.
<code>(NSPoint)locationOfPrintRect:(NSRect)aRect</code>	Locates the printing rectangle on the page.
<code>(NSRect)rectForPage:(int)page</code>	Provides how much of the view will print on page.
<code>(float)widthAdjustLimit</code>	Returns how much of a page can go on the next page.
<code>(id)addToPageSetup</code>	Allows you to adjust for differences in the graphics state between the printer.
<code>(id)beginPage:(int)ordinalNum label:(NSString *)aString bBox:(NSRect)pageRect fonts:(NSString *)fontNames</code>	Writes a page separator.
<code>(id)beginPageSetupRect:(NSRect)aRect placement:(NSPoint)location</code>	Writes the beginning of a page setup section.
<code>(id)beginPrologueBBox:(NSRect)boundingBox creationDate:(NSString *)dateCreated createdBy:(NSString *)anApplication fonts:(NSString *)fontNames forWhom:(NSString *)user pages:(int)numPages title:(NSString *)aTitle</code>	Writes the header for a print job.
<code>(id)beginSetup</code>	Writes the beginning of the job setup section.
<code>(id)beginTrailer</code>	Writes the beginning of the trailer for the print job.
<code>(id)drawPageBorderWithSize:(NSSize)borderSize</code>	

id)endPageSetup	Writes the end of a page setup section.
id)endPage	Writes the end of a page.
id)endTrailer	Writes the end of the trailer.