Application Kit Reference Update

Cocoa

2007-07-18

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Introduction to Application Kit Reference Update

This document summarizes the symbols that have been added to the Application Kit framework. The full reference documentation notes in what version a symbol was introduced, but sometimes it's useful to see only the new symbols for a given release.

If you are not familiar with this framework you should refer to the complete framework reference documentation.

Organization of This Document

Symbols are grouped by class or protocol for Objective-C and by header file for C. For each symbol there is a link to complete documentation, if available, and a brief description, if available.

See Also

For reference documentation on this framework, see *Application Kit Framework Reference*.

10.5 Symbol Changes

This article lists the symbols added to AppKit.framework in Mac OS X v10.5.

Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

NSActionCell

Complete reference information is available in the NSActionCell reference.

Instance Methods

integerValue	Returns the receiver's value as a 64-bit compatible integer after validating any editing of cell content.

NSAlert

Complete reference information is available in the NSAlert reference.

Instance Methods

accessoryView	Returns the receiver's accessory view.
layout	Specifies that the receiver must do immediate layout instead of lazily just before display.
setAccessoryView:	Sets the receiver's accessory view.
setShowsSuppressionButton:	Specifies whether the receiver includes a suppression checkbox.
showsSuppressionButton	Indicates whether the receiver shows a suppression button.
suppressionButton	Returns the receiver's suppression checkbox.

NSAnimationContext (New)

Complete reference information is available in the NSAnimationContext reference.

Class Methods

beginGrouping	Creates a new animation grouping.
currentContext	Returns the current animation context.
endGrouping	Ends the current animation grouping.

Instance Methods

duration	Returns the duration used when animating object properties that support animation.
setDuration:	Sets the duration used by animations created as a result of setting new values for an animatable property.

NSApplication

Complete reference information is available in the NSApplication reference.

Instance Methods

dockTile	Returns the application’s Dock tile.

NSArrayController

Complete reference information is available in the NSArrayController reference.

Instance Methods

automaticallyRearrangesObjects	Returns a Boolean that indicates if the receiver automatically rearranges its content to correspond to the current sort descriptors and filter predicates.
automaticRearrangementKeyPaths	Returns an array of key paths that trigger automatic content sorting or filtering.
didChangeArrangementCriteria	Invoked to inform the controller that the arrangement criteria has changed..

`setAutomaticallyRearrangesObjects:`	Sets whether or not the receiver automatically rearranges its content to correspond to the current sort descriptors and filter predicates.

NSAttributedString

Complete reference information is available in the `NSAttributedString` reference.

Class Methods

`textTypes`	Returns an array of UTI strings identifying the file types supported by the receiver, either directly or through a user-installed filter service.
`textUnfilteredTypes`	Returns an array of UTI strings identifying the file types supported directly by the receiver.

Instance Methods

`URLAtIndex:effectiveRange:`	Returns a URL, either from a link attribute or from text at the given location that appears to be a URL string, for use in automatic link detection.

NSBezierPath

Complete reference information is available in the `NSBezierPath` reference.

Class Methods

`bezierPathWithRoundedRect:xRadius:yRadius:`	Creates and returns a new NSBezierPath object initialized with a rounded rectangular path.

Instance Methods

`appendBezierPathWithRoundedRect:xRadius:yRadius:`	Appends a rounded rectangular path to the receiver’s path.

NSBitmapImageRep

Complete reference information is available in the `NSBitmapImageRep` reference.

Instance Methods

CGImage	Returns a Core Graphics image object from the receiver's current bitmap data.
initWithCGImage:	Returns an NSBitmapImageRep object created from a Core Graphics image object.
initWithCIImage:	Returns an NSBitmapImageRep object created from a Core Image object.

NSBox

Complete reference information is available in the NSBox reference.

Instance Methods

borderColor	Returns the color of the receiver's border when the receiver is a custom box with a simple line border.
borderWidth	Returns the width of the receiver's border when the receiver is a custom box with a simple line border.
cornerRadius	Returns the radius of the receiver's corners when the receiver is a custom box with a simple line border.
fillColor	Returns the color of the receiver's background when the receiver is a custom box with a simple line border.
isTransparent	Indicates whether the receiver is transparent.
setBorderColor:	Specifies the receiver's border color.
setBorderWidth:	Specifies the receiver's border width.
setCornerRadius:	Specifies the receiver's corner radius.
setFillColor:	Specifies the receiver's fill color.
setTransparent:	Specifies whether the receiver is transparent.

NSBrowser

Complete reference information is available in the NSBrowser reference.

Instance Methods

allowsTypeSelect	Indicates whether the receiver allows keystroke-based selection (type select).

backgroundColor	Provides the receiver's background color. Default: [NSColor whiteColor].
canDragRowsWithIndexes:inColumn:withEvent:	Indicates whether the receiver can attempt to initiate a drag of the given rows for the given event.
draggingImageForRowsWithIndexes:inColumn:withEvent:offset:	Provides an image to represent dragged rows during a drag operation on the receiver.
selectedRowIndexesInColumn:	Provides the indexes of the selected rows in a given column of the receiver.
selectRowIndexes:inColumn:	Specifies the selected rows in a given column of the receiver.
setAllowsTypeSelect:	Specifies whether the receiver allows keystroke-based selection.
setBackgroundColor:	Specifies the receiver's background color.
setDraggingSourceOperationMask:forLocal:	Specifies the drag-operation mask for dragging operations with local or external destinations.

Delegate Methods

browser:acceptDrop:atRow:column:dropOperation:	Sent to the delegate during a dragging session to determine whether to accept the drop. Required for a browser to be a drag destination.
browser:canDragRowsWithIndexes:inColumn:withEvent:	Sent to the delegate to determine whether the browser can attempt to initiate a drag of the given rows for the given event.
browser:draggingImageForRowsWithIndexes:inColumn:withEvent:offset:	Sent to the delegate to obtain an image to represent dragged rows during a drag operation on a browser.
browser:namesOfPromisedFilesDroppedAtDestination:forDraggedRowsWithIndexes:inColumn:	Implements file-promise drag operations.
browser:nextTypeSelectMatchFromRow:toRow:inColumn:forString:	Sent to the delegate to customize a browser's keyboard-based selection (type select) behavior.
browser:shouldShowCellExpansionForRow:column:	Determines whether an expansion tooltip appears for a cell at the given row in the given column.

`browser:shouldTypeSelectForEvent: withCurrentSearchString:`	Sent to the delegate to determine whether keyboard-based selection (type select) for a given event and search string should proceed.
`browser:typeSelectStringForRow:inColumn:`	Sent to the delegate to get the keyboard-based selection (type select) string for a given row and column.
`browser:validateDrop:proposedRow: column:dropOperation:`	Sent to the delegate during a dragging session to determine whether to accept a drop, and to obtain the drop location. Required for a browser to be a drag destination.
`browser:writeRowsWithIndexes: inColumn:toPasteboard:`	Determines whether a drag operation can proceed. Required for a browser to be a drag source.

NSButtonCell

Complete reference information is available in the `NSButtonCell` reference.

Instance Methods

`imageScaling`	Returns the scale factor for the receiver's image.
`setImageScaling:`	Sets the scale factor for the receiver's image.

NSCell

Complete reference information is available in the `NSCell` reference.

Instance Methods

`backgroundStyle`	Returns the background style for the receiver.
`drawWithExpansionFrame:inView:`	Instructs the receiver to draw in an expansion frame.
`expansionFrameWithFrame:inView:`	Returns the expansion cell frame for the receiver.
`hitTestForEvent:inRect:ofView:`	Returns hit testing information for the receiver.
`integerValue`	Returns the receiver's value as an NSInteger.
`interiorBackgroundStyle`	Returns the interior background style for the receiver.
`setBackgroundStyle:`	Sets the background style for the receiver.

setIntegerValue:	Sets the value of the receiver using an NSInteger.
setTruncatesLastVisibleLine:	Sets whether the receiver truncates and adds the ellipsis character to the last visible line if the text doesn't fit into the cell bounds.
takeIntegerValueFrom:	Sets the value of the receiver's cell to an integer value obtained from the specified object.
truncatesLastVisibleLine	Returns a Boolean value indicating whether the receiver truncates and adds the ellipsis character to the last visible line if the text doesn't fit into the cell bounds.
wantsNotificationForMarkedText	Returns a Boolean value that indicates whether the field editor initiated by the receiver should post text change notifications.

NSCollectionView (New)

Complete reference information is available in the NSCollectionView reference.

Instance Methods

allowsMultipleSelection	Returns a Boolean value that indicates whether the receiver allows the user to select more than one item at a time.
backgroundColors	Return the receiver's background colors.
content	Returns the receiver's content object.
isFirstResponder	Returns whether the receiver is the first responder.
isSelectable	Returns a Boolean value that indicates whether the receiver allows the user to select items, NO if it doesn't.
itemPrototype	Returns the receiver's collection view item prototype.
maxItemSize	Returns the maximum size used to display individual collection view items in the grid
maxNumberOfColumns	Returns the maximum number of columns the receiver will display.
maxNumberOfRows	Returns the maximum number of rows the receiver will display.
minItemSize	Returns the minimum size used to display individual collection view items in the grid.
newItemForRepresentedObject:	Returns the collection view item that is used for the specified object.
selectionIndexes	Returns an index set containing the indexes of the receiver's currently selected objects in the content array.
setAllowsMultipleSelection:	Controls whether the user can select multiple items at a time.

`setBackgroundColors:`	Sets the receiver's background colors to the specified array of colors.
`setContent:`	Sets the receiver’s content array.
`setItemPrototype:`	Sets the reciever’s item prototype to the specified collection view item.
`setMaxItemSize:`	Sets the maximum size used to display individual collection view items in the grid.
`setMaxNumberOfColumns:`	Sets the maximum number of columns the receiver will display
`setMaxNumberOfRows:`	Sets the maximum number of rows the receiver will display.
`setMinItemSize:`	Sets the minimum size used to display individual layout items in the grid.
`setSelectable:`	Controls whether the receiver allows the user to select items.
`setSelectionIndexes:`	Sets the receiver’s selection using the specified indexes.

NSCollectionViewItem (New)

Complete reference information is available in the `NSCollectionViewItem` reference.

Instance Methods

`collectionView`	Returns the receiver’s collection view.
`isSelected`	Returns the selection state of the receiver.
`representedObject`	Returns the receiver’s represented object.
`setRepresentedObject:`	Sets the receiver’s represented object to the specified model object.
`setSelected:`	Sets the selection state of the receiver.
`setView:`	Sets the view the receiver uses to display its represented object.
`view`	Returns the view the receiver uses to display its represented object.

NSColorPicker

Complete reference information is available in the `NSColorPicker` reference.

Instance Methods

`buttonToolTip`	Returns the tool tip to be shown when the mouse cursor is over the receiver's button image.
`minContentSize`	Returns the minimum content size for the receiver.

NSColorSpace

Complete reference information is available in the `NSColorSpace` reference.

Class Methods

`adobeRGB1998ColorSpace`	Returns an NSColorSpace object representing an Adobe RGB (1998) color space.
`sRGBColorSpace`	Returns an NSColorSpace object representing an sRGB color space.

Instance Methods

`CGColorSpace`	Returns a Core Graphics color-space object that represents a color space equivalent to the receiver's.
`initWithCGColorSpace:`	Initializes and returns an NSColorSpace object initialized from a Core Graphics color-space object.

NSControl

Complete reference information is available in the `NSControl` reference.

Instance Methods

`integerValue`	Returns the value of the receiver's cell as an NSInteger value.
`setIntegerValue:`	Sets the value of the receiver's cell using an NSInteger value.
`takeIntegerValueFrom:`	Sets the value of the receiver's cell to an NSInteger value obtained from the specified object.

NSDictionaryController (New)

Complete reference information is available in the `NSDictionaryController` reference.

Instance Methods

excludedKeys	Returns an array containing the key names that are never displayed in the user interface items bound to the receiver.
includedKeys	Returns an array containing the key names that are represented by a key-value pair, even if they are not present in the receiver's content dictionary.
initialKey	Returns the string used as the initial key name for a newly inserted item.
initialValue	Returns the string used as the initial value for a newly inserted item.
localizedKeyDictionary	Returns the receiver's localization dictionary.
localizedKeyTable	Returns the strings file used to localize key names.
newObject	Creates and returns a new key-value pair to represent an entry in the content dictionary.
setExcludedKeys:	Sets the key names that are never displayed in the user interface items bound to the receiver.
setIncludedKeys:	Sets the key names that are represented by a key-value pair, even if they are not present in the receiver's content dictionary.
setInitialKey:	Sets the string used as the initial key name for a newly inserted item.
setInitialValue:	Sets the string used as the initial value for a newly inserted item.
setLocalizedKeyDictionary:	Sets the localized key names that are displayed by the receiver in place of the key names.
setLocalizedKeyTable:	Specifies the strings file used to localize key names.

NSDockTile (New)

Complete reference information is available in the NSDockTile reference.

Instance Methods

badgeLabel	Returns the tile's current badge label.
contentView	Returns the view used to draw the dock tile contents.
display	Redraws the dock tile's content.
owner	Returns the object represented by the dock tile.
setBadgeLabel:	Sets the string to be displayed in the tile's badging area.

setContentView:	Sets the view to use for drawing the dock tile contents.
setShowsApplicationBadge:	Sets whether the tile should be badged with the application's icon.
showsApplicationBadge	Returns a Boolean value indicating whether the tile is badged with the application's icon.
size	Returns the size of the tile.

NSDocument

Complete reference information is available in the NSDocument reference.

Instance Methods

fileNameExtensionForType:saveOperation:	Returns a filename extension that can be appended to a base filename, for a specified file type and kind of save operation.

NSEvent

Complete reference information is available in the NSEvent reference.

Class Methods

eventWithCGEvent:	Creates and returns an event object that is based on a Core Graphics type of event.
eventWithEventRef:	Creates an event object that is based on a Carbon type of event.
isMouseCoalescingEnabled	Indicates whether mouse-movement event coalescing is enabled.
setMouseCoalescingEnabled:	Controls whether mouse-movement event coalescing is enabled.

Instance Methods

CGEvent	Returns a Core Graphics event object corresponding to the receiver.
eventRef	Returns the Carbon type associated with the receiver for representing an event.
trackingArea	Returns the NSTrackingArea object that generated the event represented by the receiver.

NSFontDescriptor

Complete reference information is available in the NSFontDescriptor reference.

Instance Methods

matchingFontDescriptorWithMandatoryKeys:	Returns a normalized font descriptor whose specified attributes match those of the receiver.

NSFontManager

Complete reference information is available in the NSFontManager reference.

Instance Methods

convertFontTraits:	Converts font traits to a new traits mask value.
currentFontAction	Returns the current font conversion action.
setTarget:	Sets the target for the sendAction method.
target	Returns the target for the sendAction method.

NSGradient (New)

Complete reference information is available in the NSGradient reference.

Instance Methods

colorSpace	Returns the color space of the colors associated with the receiver.
drawFromCenter:radius:toCenter:radius:options:	Draws a radial gradient between the specified circles.
drawFromPoint:toPoint:options:	Draws a linear gradient between the specified start and end points.
drawInBezierPath:angle:	Fills the specified path with a linear gradient.
drawInBezierPath:relativeCenterPosition:	Draws a radial gradient starting at the center point of the specified path.
drawInRect:angle:	Fills the specified rectangle with a linear gradient.

drawInRect:relativeCenterPosition:	Draws a radial gradient starting at the center of the specified rectangle.
getColor:location:atIndex:	Returns information about the color stop at the specified index in the receiver's color array.
initWithColors:	Initializes a newly allocated gradient object with an array of colors.
initWithColors:atLocations:colorSpace:	Initializes a newly allocated gradient object with the specified colors, color locations, and color space.
initWithColorsAndLocations:	Initializes a newly allocated gradient object with a comma-separated list of arguments.
initWithStartingColor:endingColor:	Initializes a newly allocated gradient object with two colors.
interpolatedColorAtLocation:	Returns the color of the rendered gradient at the specified relative location.
numberOfColorStops	Returns the number of color stops associated with the receiver.

NSGraphicsContext

Complete reference information is available in the NSGraphicsContext reference.

Instance Methods

colorRenderingIntent	Returns the current rendering intent in the receiver's graphics state.
setColorRenderingIntent:	Sets the rendering intent in the receiver's graphics state.

NSImage

Complete reference information is available in the NSImage reference.

Class Methods

imageTypes	Returns an array of UTI strings identifying the image types supported by the registered NSImageRep objects, either directly or through a user-installed filter service.

`imageUnfilteredTypes`	Returns an array of UTI strings identifying the image types supported directly by the registered NSImageRep objects.

Instance Methods

`alignmentRect`	Returns alignment metadata that your code can use to position the image during layout.
`initWithIconRef:`	Initializes the image object with a Carbon-style icon resource.
`isTemplate`	Returns a Boolean value indicating whether the image is a template image.
`setAlignmentRect:`	Sets the alignment metadata that your code can use to position the image during layout.
`setTemplate:`	Sets whether the image represents a template image.

NSImageRep

Complete reference information is available in the `NSImageRep` reference.

Class Methods

`imageRepClassForType:`	Returns the NSImageRep subclass that handles image data for the specified UTI.
`imageTypes`	Returns an array of UTI strings identifying the image types supported by the receiver, either directly or through a user-installed filter service.
`imageUnfilteredTypes`	Returns an array of UTI strings identifying the image types supported directly by the receiver.

NSLayoutManager

Complete reference information is available in the `NSLayoutManager` reference.

Instance Methods

`addTemporaryAttribute:value:forCharacterRange:`	Adds a temporary attribute with the given name and value to the characters in the specified range.
`allowsNonContiguousLayout`	Indicates whether noncontiguous layout is enabled or disabled.

`attributedString`	Returns the text storage object from which the NSGlyphGenerator object procures characters for glyph generation.
`defaultBaselineOffsetForFont:`	Returns the default baseline offset specified by the layout manager's typesetter behavior for the given font.
`ensureGlyphsForCharacterRange:`	Forces the receiver to generate glyphs for the specified character range, if it has not already done so.
`ensureGlyphsForGlyphRange:`	Forces the receiver to generate glyphs for the specified glyph range, if it has not already done so.
`ensureLayoutForBoundingRect:inTextContainer:`	Forces the receiver to perform layout for the specified area in the specified text container, if it has not already done so.
`ensureLayoutForCharacterRange:`	Forces the receiver to perform layout for the specified character range, if it has not already done so.
`ensureLayoutForGlyphRange:`	Forces the receiver to perform layout for the specified glyph range, if it has not already done so.
`ensureLayoutForTextContainer:`	Forces the receiver to perform layout for the specified text container, if it has not already done so.
`getLineFragmentInsertionPointsForCharacterAtIndex:alternatePositions:inDisplayOrder:positions:characterIndexes:`	Returns insertion points in bulk for a given line fragment.
`glyphIndexForCharacterAtIndex:`	Returns the index of the first glyph associated with the character at the specified index.
`hasNonContiguousLayout`	Indicates whether the layout manager currently has any areas of noncontiguous layout.
`insertGlyphs:length:forStartingGlyphAtIndex:characterIndex:`	Inserts the given glyphs into the glyph cache at the given index and maps them to characters beginning at the given character index.
`invalidateGlyphsOnLayoutInvalidationForGlyphRange:`	Specifies explicitly when portions of the glyph stream depend on layout.

invalidateLayoutForCharacterRange: actualCharacterRange:	Invalidates the layout information for the glyphs mapped to the given range of characters.
layoutOptions	Returns the layout manager's current layout options.
setAllowsNonContiguousLayout:	Enables or disables noncontiguous layout.
setLocations:startingGlyphIndexes: count:forGlyphRange:	Sets locations for many glyph ranges at once.
setUsesFontLeading:	Specifies whether or not the receiver uses the leading provided in the font.
temporaryAttribute:atCharacterIndex: effectiveRange:	Returns the value for the temporary attribute with a given name of the character at a given index, and by reference the range over which the attribute applies.
temporaryAttribute:atCharacterIndex: longestEffectiveRange:inRange:	Returns the value for the temporary attribute with a given name of the character at a given index, and by reference the maximum range over which the attribute applies.
temporaryAttributesAtCharacterIndex: longestEffectiveRange:inRange:	Returns the temporary attributes for the character at a given index, and by reference the maximum range over which the attributes apply.
usesFontLeading	Indicates whether the receiver uses the leading provided in the font.

Delegate Methods

layoutManager:shouldUseTemporaryAttributes: forDrawingToScreen:atCharacterIndex: effectiveRange:	Sent when the layout manager is drawing and needs to decide whether or not to use temporary attributes.

NSMenu

Complete reference information is available in the NSMenu reference.

Instance Methods

cancelTracking	Dismisses the menu and ends all menu tracking.

highlightedItem	Returns the highlighted item in the receiver.
setShowsStateColumn:	Sets whether the receiver displays the state column.
showsStateColumn	Returns a Boolean value that indicates whether the receiver displays the state column.

Delegate Methods

menu:willHighlightItem:	Called to indicates that a menu is about to highlight a given item.
menuDidClose:	Sent after a menu closed.
menuWillOpen:	Sent when a menu is about to open.

NSMenuItem

Complete reference information is available in the NSMenuItem reference.

Instance Methods

isHidden	Returns a Boolean value that indicates whether the receiver is hidden.
isHiddenOrHasHiddenAncestor	Returns a Boolean value that indicates whether the receiver or any of its superitems is hidden.
isHighlighted	Returns a Boolean value that indicates whether the receiver should be drawn highlighted.
setHidden:	Sets whether the receiver is hidden.
setView:	Sets the content view for the receiver.
view	Returns the view for the receiver.

NSObject

Complete reference information is available in the NSObject reference.

Instance Methods

isExplicitlyIncluded	Specifies whether the receiver's key name is an included key.
key	Returns the receiver's key name.

`localizedKey`	Returns the receiver's localized key name.
`optionDescriptionsForBinding:`	Returns an array describing the options for the specified binding.
`ruleEditorRowsDidChange:`	Notifies the receiver that a rule editor's rows changed.
`setKey:`	Sets the key name for the receiver.
`setLocalizedKey:`	Sets the localized key name for the receiver.
`setValue:`	Sets the receiver's value.
`value`	Returns the receiver's value.

NSObjectController

Complete reference information is available in the `NSObjectController` reference.

Instance Methods

`defaultFetchRequest`	Returns the default fetch request used by the receiver.
`setUsesLazyFetching:`	Sets whether the receiver uses lazy fetching.
`usesLazyFetching`	Returns a Boolean indicating whether the receiver uses lazy fetching.
`validateUserInterfaceItem:`	Returns whether the receiver can handle the action method for a user interface item.

NSOutlineView

Complete reference information is available in the `NSOutlineView` reference.

Instance Methods

`frameOfOutlineCellAtRow:`	Returns the frame of the outline cell for a given row.
`parentForItem:`	Returns the parent for a given item.

Delegate Methods

`outlineView:dataCellForTableColumn:item:`	Returns the cell to use in a given column for a given item.

`outlineView:isGroupItem:`	Returns a Boolean that indicates whether a given row should be drawn in the “group row” style.
`outlineView:nextTypeSelectMatchFromItem: toItem:forString:`	Returns the first item that matches the searchString from within the range of startItem to endItem
`outlineView:selectionIndexesForProposedSelection:`	Invoked to allow the delegate to modify the proposed selection.
`outlineView:shouldShowCellExpansionForTableColumn: item:`	Returns a Boolean value that indicates whether an expansion tooltip should appear in a given column for a given item.
`outlineView:shouldTrackCell:forTableColumn:item:`	Returns a Boolean value that indicates whether a given cell should be tracked.
`outlineView:shouldTypeSelectForEvent: withCurrentSearchString:`	Returns a Boolean value that indicates whether type select should proceed for a given event and search string.
`outlineView:typeSelectStringForTableColumn:item:`	Returns the string that is used for type selection for a given column and item.

NSPageLayout

Complete reference information is available in the `NSPageLayout` reference.

Instance Methods

`accessoryControllers`	Returns an array of accessory view controllers belonging to the receiver.
`addAccessoryController:`	Adds the given controller of an accessory view to be presented in the page setup panel.
`removeAccessoryController:`	Removes the given controller of an accessory view.

NSPathCell (New)

Complete reference information is available in the `NSPathCell` reference.

Class Methods

`pathComponentCellClass`	Returns the class used to create pathComponentCell objects when automatically filling up the control.

Instance Methods

`allowedTypes`	Returns the component types allowed in the path when the cell is editable.
`backgroundColor`	Returns the current background color of the receiver.
`clickedPathComponentCell`	Returns the clicked cell.
`delegate`	Returns the receiver's delegate.
`doubleAction`	Returns the receiver's double-click action method.
`mouseEntered:withFrame:inView:`	Displays the cell component over which the mouse is hovering.
`mouseExited:withFrame:inView:`	Hides the cell component over which the mouse is hovering.
`pathComponentCellAtPoint:withFrame:inView:`	Returns the cell located at the given point within the given frame of the given view.
`pathComponentCells`	Returns an array of the NSPathComponentCell objects currently being displayed.
`pathStyle`	Returns the receiver's path style.
`placeholderAttributedString`	Returns the placeholder attributed string.
`placeholderString`	Returns the placeholder string.
`rectOfPathComponentCell:withFrame:inView:`	Returns the current rectangle being displayed for a given path component cell, with respect to a given frame in a given view.
`setAllowedTypes:`	Sets the component types allowed in the path when the cell is editable.
`setBackgroundColor:`	Sets the receiver's background color.
`setControlSize:`	Sets the receiver's control size.
`setDelegate:`	Sets the receiver's delegate.
`setDoubleAction:`	Sets the receiver's double-click action.
`setObjectValue:`	Sets the receiver's object value.

`setPathComponentCells:`	Sets the array of NSPathComponentCell objects currently being displayed.
`setPathStyle:`	Sets the receiver's path style.
`setPlaceholderAttributedString:`	Sets the value of the placeholder attributed string.
`setPlaceholderString:`	Sets the value of the placeholder string.
`setURL:`	Sets the value of the path displayed by the receiver.
`URL`	Returns the path displayed by the receiver.

NSPathComponentCell (New)

Complete reference information is available in the `NSPathComponentCell` reference.

Instance Methods

`image`	Returns the image displayed for this component cell.
`setImage:`	Sets the image displayed for this component cell.
`setURL:`	Sets the value of the portion of the path from the root through the component represented by the receiver.
`URL`	Returns the portion of the path from the root through the component represented by the receiver.

NSPathControl (New)

Complete reference information is available in the `NSPathControl` reference.

Instance Methods

`backgroundColor`	Returns the current background color of the receiver.
`clickedPathComponentCell`	Returns the clicked cell, or nil, if no cell was clicked.
`delegate`	Returns the receiver's delegate.
`doubleAction`	Returns the receiver's double-click action method.
`pathComponentCells`	Returns an array of the NSPathComponentCell objects currently being displayed.

`pathStyle`	Returns the receiver’s path style.
`setBackgroundColor:`	Sets the receiver’s background color.
`setDelegate:`	Sets the receiver’s delegate.
`setDoubleAction:`	Sets the receiver’s double-click action.
`setDraggingSourceOperationMask:forLocal:`	Configures the default value returned from draggingSourceOperationMaskForLocal: .
`setPathComponentCells:`	Sets the array of NSPathComponentCell objects currently being displayed.
`setPathStyle:`	Sets the receiver’s path style.
`setURL:`	Sets the path value displayed by the receiver.
`URL`	Returns the path value displayed by the receiver.

NSPersistentDocument

Complete reference information is available in the `NSPersistentDocument` reference.

Instance Methods

`configurePersistentStoreCoordinatorForURL:ofType:`
`modelConfiguration:storeOptions:error:`
	Configures the receiver’s persistent store coordinator with the appropriate stores for a given URL.

NSPredicateEditor (New)

Complete reference information is available in the `NSPredicateEditor` reference.

Instance Methods

`rowTemplates`	Returns the row templates for the receiver.
`setRowTemplates:`	Sets the row templates for the receiver.

NSPredicateEditorRowTemplate (New)

Complete reference information is available in the `NSPredicateEditorRowTemplate` reference.

Class Methods

`templatesWithAttributeKeyPaths:`
`inEntityDescription:`
	Returns an array of predicate templates for the given attribute key paths for a given entity.

Instance Methods

`compoundTypes`	Returns the compound predicate types for the receiver.
`displayableSubpredicatesOfPredicate:`	Returns the subpredicates that should be made sub-rows of a given predicate.
`initWithCompoundTypes:`	Initializes and returns a row template suitable for displaying compound predicates.

`initWithLeftExpressions:`
`rightExpressionAttributeType:modifier:operators:`
`options:`
	Initializes and returns a "pop-up-pop-up-view"-style row template.

`initWithLeftExpressions:rightExpressions:`
`modifier:operators:options:`
	Initializes and returns a "pop-up-pop-up-pop-up"-style row template.

`leftExpressions`	Returns the left hand expressions for the receiver.
`matchForPredicate:`	Returns a positive number if the receiver can represent a given predicate, and 0 if it cannot.
`modifier`	Returns the comparison predicate modifier for the receiver.
`operators`	Returns the array of operators for the receiver.
`options`	Returns the comparison predicate options for the receiver.
`predicateWithSubpredicates:`	Returns the predicate represented by the receiver's views' values and the given sub-predicates.
`rightExpressionAttributeType`	Returns the attribute type of the receiver's right expression.
`rightExpressions`	Returns the right hand expressions for the receiver.
`setPredicate:`	Sets the value of the views according to the given predicate.

`templateViews`	Returns the views for the receiver.

NSPrintInfo

Complete reference information is available in the `NSPrintInfo` reference.

Instance Methods

`PMPageFormat`	Returns a Core Printing object configured with the receiver's page format information.
`PMPrintSession`	Returns a Core Printing object configured with the receiver's session information.
`PMPrintSettings`	Returns a Core Printing object configured with the receiver's print settings information
`printSettings`	Returns a mutable dictionary containing the print settings from Core Printing.
`updateFromPMPageFormat`	Synchronizes the receiver's page format information with information from its associated PMPageFormat object.
`updateFromPMPrintSettings`	Synchronizes the receiver's print settings information with information from its associated PMPrintSettings object.

NSPrintOperation

Complete reference information is available in the `NSPrintOperation` reference.

Instance Methods

`jobTitle`	Returns the title of the print job.
`pageRange`	Returns the range of pages associated with the print operation.
`setJobTitle:`	Assigns a custom title to the print job.

NSPrintPanel

Complete reference information is available in the `NSPrintPanel` reference.

Instance Methods

accessoryControllers	Returns the array of controller objects used to manage the Print panel's accessory views.
addAccessoryController:	Adds a custom controller to the Print panel to manage an accessory view.
defaultButtonTitle	Returns the title of the Print panel's default button.
helpAnchor	Returns the HTML help anchor associated with the Print panel.
options	Returns the current configuration options for the Print panel.
printInfo	Returns the printing information associated with the running Print panel.
removeAccessoryController:	Removes the specified controller and accessory view from the Print panel.
runModalWithPrintInfo:	Displays the receiver's Print panel and runs the modal loop using the specified printing information.
setDefaultButtonTitle:	Sets the title of the Print panel's default button.
setHelpAnchor:	Sets the HTML help anchor for the print panel.
setOptions:	Sets the configuration options for the Print panel.

NSResponder

Complete reference information is available in the NSResponder reference.

Instance Methods

cursorUpdate:	Informs the receiver that the mouse cursor has moved into a cursor rectangle.

NSRuleEditor (New)

Complete reference information is available in the NSRuleEditor reference.

Instance Methods

addRow:	Adds a row to the receiver.
canRemoveAllRows	Returns a Boolean value that indicates whether all the rows can be removed.

`criteriaForRow:`	Returns the currently chosen items for a given row.
`criteriaKeyPath`	Returns the criteria key path.
`delegate`	Returns the receiver's delegate.
`displayValuesForRow:`	Returns the chosen values for a given row.
`displayValuesKeyPath`	Returns the display values key path.
`formattingDictionary`	Returns the formatting dictionary for the receiver.
`formattingStringsFilename`	Returns the name of the strings file for the receiver.
`insertRowAtIndex:withType:asSubrowOfRow:animate:`	Adds a new row of a given type at a given location.
`isEditable`	Returns a Boolean value that indicates whether the receiver is editable.
`nestingMode`	Returns the nesting mode for the receiver.
`numberOfRows`	Returns the number of rows in the receiver.
`parentRowForRow:`	Returns the index of the parent of a given row.
`predicate`	Returns the predicate for the receiver.
`predicateForRow:`	Returns the predicate for a given row.
`reloadCriteria`	Instructs the receiver to refetch criteria from its delegate.
`reloadPredicate`	Instructs the receiver to regenerate its predicate by invoking the corresponding delegate method.
`removeRowAtIndex:`	Removes the row at a given index.
`removeRowsAtIndexes:includeSubrows:`	Removes the rows at given indexes.
`rowClass`	Returns the class used to create a new row in the "rows" binding.
`rowForDisplayValue:`	Returns the index of the row containing a given value.
`rowHeight`	Returns the row height for the receiver.

`rowTypeForRow:`	Returns the type of a given row.
`rowTypeKeyPath`	Returns the key path for the row type.
`selectedRowIndexes`	Returns the indexes of the receiver's selected rows.
`selectRowIndexes:byExtendingSelection:`	Sets in the receiver the indexes of rows that are selected.
`setCanRemoveAllRows:`	Sets whether all the rows can be removed.
`setCriteria:andDisplayValues:forRowAtIndex:`	Modifies the row at a given index to contain the given items and values.
`setCriteriaKeyPath:`	Sets the criteria key path.
`setDelegate:`	Sets the receiver's delegate.
`setDisplayValuesKeyPath:`	Sets the display values key path.
`setEditable:`	Sets whether the receiver is editable.
`setFormattingDictionary:`	Sets the formatting dictionary for the receiver.
`setFormattingStringsFilename:`	Sets the name of the strings file used for formatting.
`setNestingMode:`	Sets the nesting mode for the receiver.
`setRowClass:`	Sets the class to use to create a new row in the "rows" binding.
`setRowHeight:`	Sets the row height for the receiver.
`setRowTypeKeyPath:`	Sets the key path for the row type.
`setSubrowsKeyPath:`	Set the key path for the subrows.
`subrowIndexesForRow:`	Returns the immediate subrows of a given row.
`subrowsKeyPath`	The key path for the subrows.

Delegate Methods

`ruleEditor:child:forCriterion:withRowType:`	Returns the child of a given item at a given index.
`ruleEditor:displayValueForCriterion:inRow:`	Returns the value for a given criterion.
`ruleEditor:numberOfChildrenForCriterion:withRowType:`	Returns the number of child items of a given criterion or row type.

`ruleEditor:predicatePartsForCriterion: withDisplayValue:inRow:`	Returns a dictionary representing the parts of the predicate determined by the given criterion and value.

NSScroller

Complete reference information is available in the `NSScroller` reference.

Instance Methods

`drawKnobSlotInRect:highlight:`	Draws the portion of the scroller's track, possibly including the line increment and decrement arrow buttons, that falls in the given slotRect.
`setKnobProportion:`	Sets the proportion of the knob slot the knob should fill.

NSSegmentedCell

Complete reference information is available in the `NSSegmentedCell` reference.

Instance Methods

`imageScalingForSegment:`	Returns the image scaling mode associated with the specified segment.
`interiorBackgroundStyleForSegment:`	Returns the interior background style for the specified segment.
`segmentStyle`	Returns the visual style used to display the receiver.
`setImageScaling:forSegment:`	Sets the image scaling mode for the specified segment.
`setSegmentStyle:`	Sets the visual style used to display the receiver.

NSSegmentedControl

Complete reference information is available in the `NSSegmentedControl` reference.

Instance Methods

`imageScalingForSegment:`	Returns the scaling mode used to display the specified segment's image.
`segmentStyle`	Returns the visual style used to display the receiver.

setImageScaling:forSegment:	Sets the scaling mode used to display the specified segment's image.
setSegmentStyle:	Sets the visual style used to display the receiver.

NSSound

Complete reference information is available in the NSSound reference.

Class Methods

soundUnfilteredTypes	Provides the file types the NSSound class understands.

Instance Methods

channelMapping	Provides the receiver's channel map.
currentTime	Provides the receiver's playback progress in seconds.
duration	Provides the duration of the receiver in seconds.
loops	Indicates whether the receiver restarts playback when it reaches the end of its content. Default: NO.
playbackDeviceIdentifier	Identifies the receiver's output device.
setChannelMapping:	Specifies the receiver's channel map.
setCurrentTime:	Specifies the receivers playback progress in seconds.
setLoops:	Specifies whether the receiver restarts playback when it reaches the end of its content.
setPlaybackDeviceIdentifier:	Specifies the receiver's output device.
setVolume:	Specifies the volume of the receiver.
volume	Provides the volume of the receiver.

NSSpeechSynthesizer

Complete reference information is available in the NSSpeechSynthesizer reference.

Instance Methods

addSpeechDictionary:	Registers the given speech dictionary with the receiver.

continueSpeaking	Resumes synthesis.
objectForProperty:error:	Provides the value of a receiver's property.
pauseSpeakingAtBoundary:	Pauses synthesis in progress at a given boundary.
phonemesFromText:	Provides the phoneme symbols generated by the given text.
rate	Provides the receiver's speaking rate.
setObject:forProperty:error:	Specifies the value of a receiver's property.
setRate:	Specifies the receivers speaking rate.
setVolume:	Specifies the receiver's speaking volume.
stopSpeakingAtBoundary:	Stops synthesis in progress at a given boundary.
volume	Provides the receiver's speaking volume.

Delegate Methods

speechSynthesizer:didEncounterErrorAtIndex:ofString:message:	Sent to the delegate when a speech synthesizer encounters an error in text being synthesized.
speechSynthesizer:didEncounterSyncMessage:	Sent to the delegate when a speech synthesizer encounters a synchronization error.

NSSpellChecker

Complete reference information is available in the NSSpellChecker reference.

Instance Methods

availableLanguages	Provides the languages preferred by the user.
checkGrammarOfString:startingAt:language:wrap:inSpellDocumentWithTag:details:	Initiates a grammatical analysis of a given string.
forgetWord:	
hasLearnedWord:	Indicates whether the spell checker has learned a given word.
learnWord:	
unlearnWord:	Tells the spell checker to unlearn a given word.
updateSpellingPanelWithGrammarString:detail:	Specifies a grammar-analysis detail to highlight in the Spelling panel.

NSSplitView

Complete reference information is available in the `NSSplitView` reference.

Instance Methods

`autosaveName`	Returns the name under which receiver's divider position is automatically saved.
`dividerColor`	Return the color of the dividers that the split view is drawing between subviews.
`dividerStyle`	Returns the style of the divider drawn between subviews.
`maxPossiblePositionOfDividerAtIndex:`	Returns the maximum possible position of the divider at the specified index.
`minPossiblePositionOfDividerAtIndex:`	Returns the minimum possible position of the divider at the specified index.
`setAutosaveName:`	Sets the name under which receiver's divider position is automatically saved.
`setDividerStyle:`	Sets the style of divider drawn between subviews.
`setPosition:ofDividerAtIndex:`	Sets the position of the divider at the specified index.

Delegate Methods

`splitView:additionalEffectiveRectOfDividerAtIndex:`	Allows the delegate to return an additional rectangle in which mouse clicks will initiate divider dragging.
`splitView:effectiveRect:forDrawnRect:ofDividerAtIndex:`	Allows the delegate to modify the rectangle in which mouse clicks initiate divider dragging.
`splitView:shouldCollapseSubview:forDoubleClickOnDividerAtIndex:`	Invoked to allow a delegate to determine if a subview should collapse in response to a double click.
`splitView:shouldHideDividerAtIndex:`	Allows the delegate to determine whether a divider can be dragged or adjusted off the edge of the split view.

NSTableColumn

Complete reference information is available in the `NSTableColumn` reference.

Instance Methods

`headerToolTip`	Returns the tooltip string that is displayed when the cursor pauses over the header cell of the receiver.
`isHidden`	Returns a Boolean value that indicates whether the receiver is hidden.
`setHeaderToolTip:`	Sets the tooltip string that is displayed when the cursor pauses over the header cell of the receiver.
`setHidden:`	Sets whether the receiver is hidden.

NSTableView

Complete reference information is available in the `NSTableView` reference.

Instance Methods

`allowsTypeSelect`	Returns a Boolean value that indicates whether the receiver allows the user to type characters to select rows.
`columnIndexesInRect:`	Returns the indexes of the receiver's columns that intersect the specified rectangle.
`preparedCellAtColumn:row:`	Returns the fully prepared cell that the receiver will use for drawing or processing of the specified row and column.
`selectionHighlightStyle`	Returns the selection highlight style used by the receiver to indicate row and column selection.
`setAllowsTypeSelect:`	Sets whether the receiver allows the user to type characters to select rows.
`setSelectionHighlightStyle:`	Sets the selection highlight style used by the receiver to indicate row and column selection.

Delegate Methods

`tableView:dataCellForTableColumn:row:`	Invoked to allow the delegate to return a custom data cell for a specified row and column.
`tableView:isGroupRow:`	Invoked to allow the delegate to indicate that a specified row is a group row.
`tableView:nextTypeSelectMatchFromRow:toRow:forString:`	Invoked to allow the delegate to allow the delegate to modify how type selection works.

tableView:selectionIndexesForProposedSelection:	Invoked to allow the delegate to modify the proposed selection.
tableView:shouldShowCellExpansionForTableColumn:row:	Invoked to allow the delegate to control tooltip cell expansion for a specific row and column.
tableView:shouldTrackCell:forTableColumn:row:	Invoked to allow the delegate to control the tracking behavior for a specific cell.
tableView:shouldTypeSelectForEvent:withCurrentSearchString:	Invoked to allow the delegate to control type select for a specific event.
tableView:typeSelectStringForTableColumn:row:	Invoked to allow the delegate to provide an alternate text value used for type selection for a specified row and column.

NSTextFieldCell

Complete reference information is available in the NSTextFieldCell reference.

Instance Methods

allowedInputSourceLocales	Returns an array of locale identifiers representing input sources that are allowed to be enabled when the receiver has the keyboard focus.
setAllowedInputSourceLocales:	Sets an array of locale identifiers representing input sources that are allowed to be enabled when the receiver has the keyboard focus.
setWantsNotificationForMarkedText:	Directs the cell's associated field editor to post text change notifications.

NSTextView

Complete reference information is available in the NSTextView reference.

Instance Methods

allowedInputSourceLocales	Returns an array of locale identifiers representing input sources that are allowed to be enabled when the receiver has the keyboard focus.
allowsImageEditing	Indicates whether image attachments should permit editing of their images.

`characterIndexForInsertionAtPoint:`	Returns a character index appropriate for placing a zero-length selection for an insertion point associated with the mouse at the given point.
`displaysLinkToolTips`	Indicates whether the text view automatically supplies the destination of a link as a tooltip for text that has a link attribute.
`isAutomaticLinkDetectionEnabled`	Indicates whether automatic link detection is enabled.
`isAutomaticQuoteSubstitutionEnabled`	Indicates whether automatic quotation mark substitution is enabled.
`isGrammarCheckingEnabled`	Indicates whether or not grammar checking is enabled.
`setAllowedInputSourceLocales:`	Sets an array of locale identifiers representing input sources that are allowed to be enabled when the receiver has the keyboard focus.
`setAllowsImageEditing:`	Specifies whether image attachments should permit editing of their images.
`setAutomaticLinkDetectionEnabled:`	Enables or disables automatic link detection.
`setAutomaticQuoteSubstitutionEnabled:`	Enables and disables automatic quotation mark substitution.
`setDisplaysLinkToolTips:`	Enables or disables automatic display of link tooltips.
`setGrammarCheckingEnabled:`	Enables and disables grammar checking.
`setSpellingState:range:`	Sets the spelling state, which controls the display of the spelling and grammar indicators on the given text range.
`showFindIndicatorForRange:`	Causes a temporary highlighting effect to appear around the visible portion (or portions) of the specified range.
`toggleAutomaticLinkDetection:`	Changes the state of automatic link detection from enabled to disabled and vice versa.
`toggleAutomaticQuoteSubstitution:`	Changes the state of automatic quotation mark substitution from enabled to disabled and vice versa.
`toggleGrammarChecking:`	Changes the state of grammar checking from enabled to disabled and vice versa.
`toggleSmartInsertDelete:`	Changes the state of smart insert and delete from enabled to disabled and vice versa.

Delegate Methods

`textView:menu:forEvent:atIndex:`	

textView:shouldSetSpellingState:range:	Sent when the spelling state is changed.

NSToolbarItemGroup (New)

Complete reference information is available in the NSToolbarItemGroup reference.

Instance Methods

setSubitems:	Sets the subitems for the receiver.
subitems	Returns the subitems for the receiver.

NSTrackingArea (New)

Complete reference information is available in the NSTrackingArea reference.

Instance Methods

initWithRect:options:owner:userInfo:	Initializes and returns an object defining a region of a view to receive mouse-tracking events, mouse-moved events, cursor-update events, or possibly all these events.
options	Returns the options specified for the receiver.
owner	Returns the object owning the receiver, which is the recipient of mouse-tracking, mouse-movement, and cursor-update messages.
rect	Returns the rectangle defining the area encompassed by the receiver.
userInfo	Returns the dictionary containing the data associated with the receiver when it was created.

NSTreeController

Complete reference information is available in the NSTreeController reference.

Instance Methods

childrenKeyPathForNode:	Returns the key path used to find the children in the specified tree node.

`countKeyPathForNode:`	Returns the key path that provides the number of children for a specified node.
`leafKeyPathForNode:`	Returns the key path that specifies whether the node is a leaf node.
`moveNode:toIndexPath:`	Moves the specified tree node to the new index path.
`moveNodes:toIndexPath:`	Moves the specified tree nodes to the new index path.
`selectedNodes`	Returns an array of the receiver's selected tree nodes.

NSTreeNode (New)

Complete reference information is available in the `NSTreeNode` reference.

Class Methods

`treeNodeWithRepresentedObject:`	Creates and returns a tree node that represents the specified object.

Instance Methods

`childNodes`	Returns an array containing receiver's child nodes.
`descendantNodeAtIndexPath:`	Returns the receiver's descendent at the specified index path.
`indexPath`	Returns the position of the receiver relative to its root parent.
`initWithRepresentedObject:`	Initializes a newly allocated tree node that represents the specified object.
`isLeaf`	Returns whether the receiver is a leaf node.
`mutableChildNodes`	Returns a mutable array that provides read-write access to the receiver's child nodes.
`parentNode`	Returns the receiver's parent node.
`representedObject`	Returns the object the tree node represents.
`sortWithSortDescriptors:recursively:`	Sorts the receiver's subtree using the values of the represented objects with the specified sort descriptors.

NSTypesetter

Complete reference information is available in the `NSTypesetter` reference.

Instance Methods

`layoutCharactersInRange:forLayoutManager: maximumNumberOfLineFragments:`	

NSView

Complete reference information is available in the `NSView` reference.

Instance Methods

`addTrackingArea:`	Adds a given tracking area to the receiver.
`alphaValue`	Returns the opacity of the receiver
`backgroundFilters`	Returns the array of CoreImage filters that are applied to the receiver's background
`compositingFilter`	Returns the CoreImage filter that is used to composite the receiver's contents with the background
`contentFilters`	Returns the array of CoreImage filters that are applied to the contents of the receiver and its sublayers.
`convertPointFromBase:`	Converts the point from the base coordinate system to the receiver's coordinate system.
`convertPointToBase:`	Converts the point from the receiver's coordinate system to the base coordinate system.
`convertRectFromBase:`	Converts the rectangle from the base coordinate system to the receiver's coordinate system.
`convertRectToBase:`	Converts the rectangle from the receiver's coordinate system to the base coordinate system.
`convertSizeFromBase:`	Converts the size from the base coordinate system to the receiver's coordinate system.
`convertSizeToBase:`	Converts the size from the receiver's coordinate system to the base coordinate system.
`enclosingMenuItem`	Returns the menu item containing the receiver or any of its superviews in the view hierarchy.
`enterFullScreenMode:withOptions:`	Sets the receiver to full screen mode.

`exitFullScreenModeWithOptions:`	Instructs the receiver to exit full screen mode.
`frameCenterRotation`	Returns the receiver's rotation about the layer's position.
`isInFullScreenMode`	Returns a Boolean value that indicates whether the receiver is in full screen mode.
`layer`	Returns the Core Animation layer that the receiver as its backing store.
`removeTrackingArea:`	Removes a given tracking area from the receiver.
`setAlphaValue:`	Sets the opacity of the receiver.
`setBackgroundFilters:`	An array of CoreImage filters that are applied to the receiver's background.
`setCompositingFilter:`	Sets a CoreImage filter that is used to composite the receiver's contents with the background.
`setContentFilters:`	Sets the array of CoreImage filters that are applied to the contents of the receiver and its sublayers.
`setFrameCenterRotation:`	Rotates the frame of the receiver about the layer's position.
`setLayer:`	Sets the Core Animation layer that the receiver uses for layer-backing to the specified layer.
`setShadow:`	Sets the shadow drawn by the receiver.
`setSubviews:`	Sets the receiver's subviews to the specified subviews.
`setWantsLayer:`	Specifes whether the receiver and its subviews use a Core Animation layer as a backing store.
`shadow`	Returns the shadow drawn by the receiver
`trackingAreas`	Returns an array of the receiver's tracking areas.
`translateRectsNeedingDisplayInRect:by:`	Translates the display rectangles by the specified delta.
`updateTrackingAreas`	Invoked automatically when the view's geometry changes such that its tracking areas need to be recalculated.
`viewDidHide`	Invoked when the receiver is hidden, either directly, or in response to an ancestor being hidden.
`viewDidUnhide`	Invoked when the receiver is unhidden, either directly, or in response to an ancestor being unhidden
`viewWillDraw`	Informs the receiver that it will be required to draw content.

wantsLayer	Returns a Boolean value that indicates whether the receiver is using a layer as it's backing store.

NSViewController (New)

Complete reference information is available in the NSViewController reference.

Instance Methods

commitEditing	Returns whether the receiver was able to commit any pending edits.
commitEditingWithDelegate:didCommitSelector:contextInfo:	Attempt to commit any currently edited results of the receiver.
discardEditing	Causes the receiver to discard any changes, restoring the previous values.
initWithNibName:bundle:	Returns an NSViewController object initialized to the nib file in the specified bundle.
loadView	Instantiate the receiver's view and set it.
nibBundle	Return the name of the nib bundle to be loaded to instantiate the receivers view.
nibName	Return the name of the nib to be loaded to instantiate the receivers view
representedObject	Returns the object whose value is being presented in the receiver's view.
setRepresentedObject:	Sets the object whose value is being presented in the receiver's view.
setTitle:	Sets the localized title of the receiver's view to the specified string.
setView:	Sets the receivers view to the specified object.
title	Returns the localized title of the receiver's view.
view	Returns the receiver's view.

NSWindow

Complete reference information is available in the NSWindow reference.

Instance Methods

autorecalculatesContentBorderThicknessForEdge:	Indicates whether the window calculates the thickness of a given border automatically.
backingLocation	Indicates the window's backing store location.
canBecomeVisibleWithoutLogin	Indicates whether the window can be displayed at the log-in window. Default: NO.
canBeVisibleOnAllSpaces	Indicates whether the window can be visible on all spaces or on only one space at a time.
collectionBehavior	Identifies the window's behavior in window collections.
contentBorderThicknessForEdge:	Indicates the thickness of a given border of the window.
dockTile	Provides the application's Dock tile.
preferredBackingLocation	Indicates the preferred location for the window's backing store.
representedURL	Provides the URL of the file the window represents.
setAutorecalculatesContentBorderThickness:forEdge:	Specifies whether the window calculates the thickness of a given border automatically.
setCanBecomeVisibleWithoutLogin:	Specifies whether the window can be displayed at the login window.
setCanBeVisibleOnAllSpaces:	Specifies whether the window can be visible on all spaces or on only one space at a time.
setCollectionBehavior:	Specifies the window's behavior in window collections.
setContentBorderThickness:forEdge:	Specifies the thickness of a given border of the window.
setPreferredBackingLocation:	Specifies the preferred location for the window's backing store.
setRepresentedURL:	Specifies the URL of the file the window represents.

`setSharingType:`	Specifies the level of access other processes have to the window’s content.
`sharingType`	Indicates the level of access other processes have to the window’s content.

Delegate Methods

`window:shouldDragDocumentWithEvent:from:withPasteboard:`	Determines whether the sender allows the user to drag the sender’s represented file’s icon from the sender’s title bar.
`window:shouldPopUpDocumentPathMenu:`	Determines whether the sender displays the title pop-up menu in response to a Command-click on the sender’s title.

NSWorkspace

Complete reference information is available in the `NSWorkspace` reference.

Instance Methods

`filenameExtension:isValidForType:`	
`localizedDescriptionForType:`	Returns the localized description for the specified Uniform Type Identifier
`preferredFilenameExtensionForType:`	Returns the preferred filename extension for the specified Uniform Type Identifier.
`type:conformsToType:`	Returns a Boolean indicating that the first Uniform Type Identifier conforms to the second Uniform Type Identifier.
`typeOfFile:error:`	Returns the uniform type identifier of the specified file, if it can be determined..

Protocols

All of the protocols with new symbols are listed alphabetically, with their new methods described.

NSAnimatablePropertyContainer (New)

Complete reference information is available in the `NSAnimatablePropertyContainer` reference.

Instance Methods

`animationForKey:`	Returns the animation that should be performed for the specified key.
`animations`	Returns the optional dictionary that maps event trigger keys to animation objects.
`animator`	Returns a proxy object for the receiver that can be used to initiate implied animation for property changes.
`defaultAnimationForKey:`	
`setAnimations:`	Sets the option dictionary that maps event trigger keys to animation objects.

NSColorPickingDefault

Complete reference information is available in the `NSColorPickingDefault` reference.

Instance Methods

`buttonToolTip`	Provides the toolbar button help tag.
`minContentSize`	Indicates the receiver's minimum content size.

NSPathCellDelegate (New)

Complete reference information is available in the `NSPathCellDelegate` reference.

Instance Methods

`pathCell:willDisplayOpenPanel:`	Implement this method to customize the Open panel shown by a pop-up–style path.
`pathCell:willPopUpMenu:`	Implement this method to customize the menu of a pop-up–style path.

NSPathControlDelegate (New)

Complete reference information is available in the `NSPathControlDelegate` reference.

Instance Methods

`pathControl:acceptDrop:`	Implement this method to accept previously validated contents dropped onto the control.
`pathControl:shouldDragPathComponentCell:withPasteboard:`	Implement this method to enable dragging from the control.
`pathControl:validateDrop:`	Implement this method to enable dragging onto the control.
`pathControl:willDisplayOpenPanel:`	Implement this method to customize the Open panel shown by a pop-up–style path.
`pathControl:willPopUpMenu:`	Implement this method to customize the menu of a pop-up–style path.

NSPrintPanelAccessorizing (New)

Complete reference information is available in the `NSPrintPanelAccessorizing` reference.

Instance Methods

`keyPathsForValuesAffectingPreview`	
`localizedSummaryItems`	

NSTextInputClient (New)

Complete reference information is available in the `NSTextInputClient` reference.

Instance Methods

`attributedString`	Returns an attributed string representing the receiver's text storage. (optional)
`attributedSubstringForProposedRange:actualRange:`	Returns an attributed string derived from the given range in the receiver's text storage.
`baselineDeltaForCharacterAtIndex:`	Returns the baseline position of a given character relative to the origin of rectangle returned by firstRectForCharacterRange: actualRange:. (optional)

`characterIndexForPoint:`	Returns the index of the character whose bounding rectangle includes the given point.
`doCommandBySelector:`	Invokes the action specified by the given selector.
`firstRectForCharacterRange:actualRange:`	Returns the first logical boundary rectangle for characters in the given range.
`fractionOfDistanceThroughGlyphForPoint:`	Returns the fraction of the distance from the left side of the character to the right side that a given point lies. (optional)
`hasMarkedText`	Returns a Boolean value indicating whether the receiver has marked text.
`insertText:replacementRange:`	Inserts the given string into the receiver, replacing the specified content.
`markedRange`	Returns the range of the marked text.
`selectedRange`	Returns the range of selected text.
`setMarkedText:selectedRange:replacementRange:`	Replaces a specified range in the receiver's text storage with the given string and sets the selection.
`unmarkText`	Unmarks the marked text.
`validAttributesForMarkedText`	Returns an array of attribute names recognized by the receiver.
`windowLevel`	Returns the window level of the receiver. (optional)

C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

AppKitErrors.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSServiceApplicationLaunchFailedError	
NSServiceApplicationNotFoundError	
NSServiceErrorMaximum	
NSServiceErrorMinimum	
NSServiceInvalidPasteboardDataError	
NSServiceMalformedServiceDictionaryError	
NSServiceMiscellaneousError	
NSServiceRequestTimedOutError	

NSAccessibility.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAccessibilityColumnCountAttribute	Number of columns in grid (NSNumber as intValue)
NSAccessibilityDisclosureTriangleRole	Disclosure triangle
NSAccessibilityGridRole	Grid
NSAccessibilityOrderedByRowAttribute	Whether grid is ordered row major if YES, column major if NO (NSNumber as boolValue)
NSAccessibilityRowCountAttribute	Number of rows in grid (NSNumber as intValue)
NSAccessibilitySelectedTextRangesAttribute	Array of NSValue (rangeValue) ranges of selected text (NSArray)
NSAccessibilityTimelineSubrole	Timeline
NSAccessibilityValueDescriptionAttribute	Text description of value (NSString)

NSAnimation.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAnimationTriggerOrderIn	The trigger that represents the action taken when a view becomes visible, either as a result of being inserted into the visible view hierarchy or the view is no longer set as hidden.
NSAnimationTriggerOrderOut	The trigger that represents the action taken when the view is either removed from the view hierarchy or is hidden.

NSApplication.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAppKitVersionNumber10_3_7	The Application Kit framework included in Mac OS X v10.3.7.
NSAppKitVersionNumber10_3_9	The Application Kit framework included in Mac OS X v10.3.9.
NSAppKitVersionNumber10_4	The Application Kit framework included in Mac OS X v10.4.

NSAttributedString.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSMarkedClauseSegmentAttributeName	NSNumber containing an integer, as an index in marked text indicating clause segments
NSOfficeOpenXMLTextDocumentType	ECMA Office Open XML text document format.
NSOpenDocumentTextDocumentType	OASIS Open Document text document format.
NSSpellingStateAttributeName	This key is available in Mac OS X v10.2 and later, but its interpretation changed in Mac OS X v10.5.
NSSpellingStateGrammarFlag	Flag for grammar issues.
NSSpellingStateSpellingFlag	Flag for spelling issues.

NSBitmapImageRep.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSImageFallbackBackgroundColor`	Specifies the background color to use when writing to an image format (such as JPEG) that doesn't support alpha. The color's alpha value is ignored. The default background color, when this property is not specified, is white. The value of the property should be an NSColor object. This constant corresponds to the kCGImageDestinationBackgroundColor constant in Quartz.

NSBox.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSBoxCustom`	Specifies that the appearance of the box is determined entirely by the by box-configuration methods, without automatically applying Apple human interface guidelines.

NSBrowser.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSBrowserDropAbove`	
`NSBrowserDropOn`	
`NSBrowserDropOperation`	This type is used to to specifying the drop type of a drag-and-drop operation. See browser:validateDrop:proposedRow: column:dropOperation for more information.

NSCell.h

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSDrawNinePartImage	Draws a nine-part tiled image.
NSDrawThreePartImage	Draws a three-part tiled image.

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSBackgroundStyle	Defines a type to specify the background style of a cell.
NSBackgroundStyleDark	The background is a dark color.
NSBackgroundStyleLight	The background is a light color.
NSBackgroundStyleLowered	The background is intended to appear lower than the content drawn on it.
NSBackgroundStyleRaised	The background is intended to appear higher than the content drawn on it.
NSCellHitContentArea	A content area in the cell.
NSCellHitEditableTextArea	An editable text area of the cell.
NSCellHitNone	An empty area, or did not hit in the cell.
NSCellHitTrackableArea	A trackable area in the cell.
NSImageScaleAxesIndependently	Scale each dimension to exactly fit destination.
NSImageScaleNone	Do not scale the image.
NSImageScaleProportionallyDown	If it is too large for the destination, scale the image down while preserving the aspect ratio.
NSImageScaleProportionallyUpOrDown	Scale the image to its maximum possible dimensions while both staying within the destination area and preserving its aspect ratio.

NSColorPanel.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSColorPanelMode`	A type defined for the enum constants specifying color panel modes.
`NSNoModeColorPanel`	Indicates no color panel mode.

NSColorSpace.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSIndexedColorSpaceModel`	An indexed color space, which identifies specified discrete colors in a color list by index number. An indexed color value (a color specification in indexed color space) consists of an index value that refers to a color in a color list.
`NSPatternColorSpaceModel`	Identifies a pattern color space, which is simply an image that is repeated over and over again in a tiled pattern.

NSDatePickerCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSTextFieldDatePickerStyle`	Provide a text field interface.

NSDocument.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSChangeRedone	A single change has been redone. For example, the built-in undo support of NSDocument passes this value whenever a document receives an NSUndoManagerDidRedoChangeNotification from its own undo manager.

NSEvent.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

buttonNumber	
deltaX	
deltaY	
reserved1	
reserved2	
subtype	

NSFontDescriptor.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSFontFeatureSelectorIdentifierKey	An NSNumber object specifying a font feature selector such as common ligature off, traditional character shape, and so on. See “Font Features“ in ATSUI Programming Guide for predefined feature selectors.
NSFontFeatureSettingsAttribute	An array of dictionaries representing non-default font feature settings.
NSFontFeatureTypeIdentifierKey	An NSNumber object specifying a font feature type such as ligature, character shape, and so on. See “Font Features“ in ATSUI Programming Guide for predefined feature types.

NSGradient.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSGradientDrawingOptions	Specifies gradient drawing options.
NSGradientDrawsAfterEndingLocation	Drawing extends beyond the gradient end point.
NSGradientDrawsBeforeStartingLocation	Drawing extends before the gradient starting point.

NSGraphicsContext.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSColorRenderingIntent	A type defined for the color rendering intent constants. (See "Color Rendering Intent Constants.")
NSColorRenderingIntentAbsoluteColorimetric	Map colors outside of the gamut of the output device to the closest possible match inside the gamut of the output device. This operation can produce a clipping effect, where two different color values in the gamut of the graphics context are mapped to the same color value in the output device's gamut. Unlike the relative colorimetric, absolute colorimetric does not modify colors inside the gamut of the output device.
NSColorRenderingIntentDefault	Use the default rendering intent for the graphics context.
NSColorRenderingIntentPerceptual	Preserve the visual relationship between colors by compressing the gamut of the graphics context to fit inside the gamut of the output device. Perceptual intent is good for photographs and other complex, detailed images.

`NSColorRenderingIntentRelativeColorimetric`	Map colors outside of the gamut of the output device to the closest possible match inside the gamut of the output device. This operation can produce a clipping effect, where two different color values in the gamut of the graphics context are mapped to the same color value in the output device's gamut. The relative colorimetric shifts all colors (including those within the gamut) to account for the difference between the white point of the graphics context and the white point of the output device.
`NSColorRenderingIntentSaturation`	Preserve the relative saturation value of the colors when converting into the gamut of the output device. The result is an image with bright, saturated colors. Saturation intent is good for reproducing images with low detail, such as presentation charts and graphs.

NSImage.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSImageNameActionTemplate`	An action menu template image.
`NSImageNameAddTemplate`	An add item template image.
`NSImageNameAdvanced`	Advanced preferences toolbar icon. Use in a preferences window only.
`NSImageNameBluetoothTemplate`	A Bluetooth template image.
`NSImageNameBonjour`	A Bonjour icon.
`NSImageNameColorPanel`	A color panel toolbar icon.
`NSImageNameColumnViewTemplate`	A column view mode template image.
`NSImageNameComputer`	A computer icon.
`NSImageNameDotMac`	A Dot Mac icon.
`NSImageNameEnterFullScreenTemplate`	An enter full-screen mode template image.
`NSImageNameEveryone`	Permissions for all users.
`NSImageNameExitFullScreenTemplate`	An exit full-screen mode template image.

NSImageNameFlowViewTemplate	A cover flow view mode template image.
NSImageNameFolderBurnable	A burnable folder icon.
NSImageNameFolderSmart	A smart folder icon.
NSImageNameFollowLinkFreestandingTemplate	A link template image. You can use this image to implement a borderless button.
NSImageNameFontPanel	A font panel toolbar icon.
NSImageNameGoLeftTemplate	A “go back” template image.
NSImageNameGoRightTemplate	A “go forward” template image.
NSImageNameIChatTheaterTemplate	An iChat Theater template image.
NSImageNameIconViewTemplate	An icon view mode template image.
NSImageNameInfo	An information toolbar icon.
NSImageNameInvalidDataFreestandingTemplate	An invalid data template image. Place this icon to the right of any fields containing invalid data. You can use this image to implement a borderless button.
NSImageNameLeftFacingTriangleTemplate	A generic left-facing triangle template image.
NSImageNameListViewTemplate	A list view mode template image.
NSImageNameLockLockedTemplate	A locked lock template image. Use to indicate locked content.
NSImageNameLockUnlockedTemplate	An unlocked lock template image. Use to indicate modifiable content that can be locked.
NSImageNameMultipleDocuments	A drag image for multiple items.
NSImageNameNetwork	A network icon.
NSImageNamePathTemplate	A path button template image.
NSImageNamePreferencesGeneral	General preferences toolbar icon. Use in a preferences window only.
NSImageNameQuickLookTemplate	A Quick Look template image.
NSImageNameRefreshFreestandingTemplate	A refresh template image. You can use this image to implement a borderless button.
NSImageNameRefreshTemplate	A refresh template image.
NSImageNameRemoveTemplate	A remove item template image.
NSImageNameRevealFreestandingTemplate	A reveal contents template image. You can use this image to implement a borderless button.

NSImageNameRightFacingTriangleTemplate	A generic right-facing triangle template image.
NSImageNameSlideshowTemplate	A slideshow template image.
NSImageNameSmartBadgeTemplate	A badge for a "smart" item.
NSImageNameStopProgressFreestandingTemplate	A stop progress template image. You can use this image to implement a borderless button.
NSImageNameStopProgressTemplate	A stop progress button template image.
NSImageNameUser	Permissions for a single user.
NSImageNameUserAccounts	User account toolbar icon. Use in a preferences window only.
NSImageNameUserGroup	Permissions for a group of users.

NSKeyValueBinding.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSContentDictionaryBinding	A dictionary used as the content dictionary.
NSContentPlacementTagBindingOption	An NSNumber object specifying the tag id of the popup menu item to replace with the content of the array.
NSExcludedKeysBinding	An array containing the key names that are never displayed in the user interface items bound to the receiver.
NSIncludedKeysBinding	An array containing the key-value pairs always represented by the receiver.
NSInitialKeyBinding	A string used as the initial key name for newly inserted items.
NSInitialValueBinding	A string used as the initial value for newly inserted items.
NSLocalizedKeyDictionaryBinding	A dictionary containing the localized key names that are displayed by the receiver in place of the key names.
NSTransparentBinding	See Cocoa Bindings Reference for more information.

NSMenu.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSMenuDidBeginTrackingNotification`	Posted when menu tracking begins.

NSOpenGL.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSOpenGLPFAAllowOfflineRenderers`	A Boolean attribute. If present, this attribute indicates that offline renderers may be used.

NSPanel.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSHUDWindowMask`	The panel is created as a transparent panel (sometimes called a “heads-up display”).

NSParagraphStyle.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`int`	

NSPasteboard.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSMultipleTextSelectionPboardType`	Multiple text selection.

NSPathCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSPathStyle`	NSPathStyle constants represent the different visual and behavioral styles an NSPathControl or NSPathCell object can have.
`NSPathStyleNavigationBar`	The navigation bar display style and behavior.
`NSPathStylePopUp`	The pop-up display style and behavior.
`NSPathStyleStandard`	The standard display style and behavior.

NSPrintPanel.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSPrintPanelAccessorySummaryItemDescriptionKey`	Used as a key to identify the current value of the accessory panel setting. The corresponding value should be an NSString object.
`NSPrintPanelAccessorySummaryItemNameKey`	Used as a key to specify the name of the accessory panel setting. The corresponding value should be an NSString object.
`NSPrintPanelOptions`	These constants are used to configure the contents of the main Print panel.

NSPrintPanelShowsCopies	The Print panel includes a field for manipulating the number of copies being printed. This field is separate from any accessory views.
NSPrintPanelShowsOrientation	The Print panel includes a control for manipulating the page orientation. This control is separate from any accessory views.
NSPrintPanelShowsPageRange	The Print panel includes a set of fields for manipulating the range of pages being printed. These fields are separate from any accessory views.
NSPrintPanelShowsPageSetupAccessory	The Print panel includes a separate accessory view for manipulating the paper size, orientation, and scaling attributes. Page setup fields that are already configured for display on the main portion of the Print panel appear there and not on this accessory panel.
NSPrintPanelShowsPaperSize	The Print panel includes a control for manipulating the paper size of the printer. This control is separate from any accessory views.
NSPrintPanelShowsPreview	The Print panel displays a built-in preview of the document contents. This option is only appropriate when the Print panel is used in conjunction with an NSPrintOperation object to print a document.
NSPrintPanelShowsScaling	The Print panel includes a control for scaling the printed output. This control is separate from any accessory views.

NSRuleEditor.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSRuleEditorNestingMode	Specifies a type for nesting modes.
NSRuleEditorNestingModeCompound	Unlimited nesting and compound rows.

NSRuleEditorNestingModeList	Allows a single list, with no nesting and no compound rows.
NSRuleEditorNestingModeSimple	One compound row at the top with subrows beneath it, and no further nesting allowed.
NSRuleEditorNestingModeSingle	Only a single row is allowed.
NSRuleEditorPredicateComparisonModifier	The corresponding value is an NSNumber object representing a NSComparisonPredicateModifier constant the of the predicate.
NSRuleEditorPredicateCompoundType	The corresponding value is an NSNumber object representing a Compound Predicate Types constant.
NSRuleEditorPredicateCustomSelector	The corresponding value is an NSString object representing a custom selector.
NSRuleEditorPredicateLeftExpression	The corresponding value is an NSExpression object representing the left expression in the predicate.
NSRuleEditorPredicateOperatorType	The corresponding value is an NSNumber object representing a NSPredicateOperatorType constant.
NSRuleEditorPredicateOptions	The corresponding value is an NSNumber object representing a NSComparisonPredicateOptions bitfield.
NSRuleEditorPredicateRightExpression	The corresponding value is an NSExpression object representing the right expression in the predicate.
NSRuleEditorRowsDidChangeNotification	This notification is posted to the default notification center whenever the view's rows change.
NSRuleEditorRowType	Specifies a type for row types.
NSRuleEditorRowTypeCompound	Specifies a compound row.
NSRuleEditorRowTypeSimple	Specifies a simple row.

NSSegmentedControl.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSSegmentStyle	The following constants specify the visual style used to display the segmented control. They are used by setSegmentStyle:.

NSSegmentStyleAutomatic	The appearance of the segmented control is automatically determined based on the type of window in which the control is displayed and the position within the window.
NSSegmentStyleCapsule	The control is displayed using the capsule style. See Figure 1 for examples.
NSSegmentStyleRounded	The control is displayed using the rounded style. See Figure 1 for examples.
NSSegmentStyleRoundRect	The control is displayed using the round rect style. See Figure 1 for examples.
NSSegmentStyleSmallSquare	The control is displayed using the small square style. See Figure 1 for examples.
NSSegmentStyleTexturedRounded	The control is displayed using the textured rounded style. See Figure 1 for examples.
NSSegmentStyleTexturedSquare	The control is displayed using the textured square style. See Figure 1 for examples.

NSSpeechSynthesizer.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSSpeechBoundary	These constants are used in stopping and pausing speech.
NSSpeechCharacterModeProperty	
NSSpeechCommandDelimiterProperty	
NSSpeechCommandPrefix	
NSSpeechCommandSuffix	
NSSpeechCurrentVoiceProperty	
NSSpeechDictionaryAbbreviations	
NSSpeechDictionaryEntryPhonemes	
NSSpeechDictionaryEntrySpelling	

NSSpeechDictionaryLocaleIdentifier	
NSSpeechDictionaryModificationDate	
NSSpeechDictionaryPronunciations	
NSSpeechErrorCount	
NSSpeechErrorNewestCharacterOffset	
NSSpeechErrorNewestCode	
NSSpeechErrorOldestCharacterOffset	
NSSpeechErrorOldestCode	
NSSpeechErrorsProperty	
NSSpeechImmediateBoundary	
NSSpeechInputModeProperty	
NSSpeechModeLiteral	
NSSpeechModeNormal	
NSSpeechModePhoneme	
NSSpeechModeText	
NSSpeechNumberModeProperty	
NSSpeechOutputToFileURLProperty	
NSSpeechPhonemeInfoExample	
NSSpeechPhonemeInfoHiliteEnd	
NSSpeechPhonemeInfoHiliteStart	
NSSpeechPhonemeInfoOpcode	
NSSpeechPhonemeInfoSymbol	
NSSpeechPhonemeSymbolsProperty	
NSSpeechPitchBaseProperty	
NSSpeechPitchModProperty	
NSSpeechRateProperty	
NSSpeechRecentSyncProperty	
NSSpeechResetProperty	

NSSpeechSentenceBoundary	
NSSpeechStatusNumberOfCharactersLeft	
NSSpeechStatusOutputBusy	
NSSpeechStatusOutputPaused	
NSSpeechStatusPhonemeCode	
NSSpeechStatusProperty	
NSSpeechSynthesizerInfoIdentifier	
NSSpeechSynthesizerInfoProperty	
NSSpeechSynthesizerInfoVersion	
NSSpeechVolumeProperty	
NSSpeechWordBoundary	
NSVoiceIndividuallySpokenCharacters	
NSVoiceLocaleIdentifier	
NSVoiceSupportedCharacters	

NSSplitView.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSSplitViewDividerStyle	These constants specify the possible divider styles used by dividerStyle and setDividerStyle:.
NSSplitViewDividerStyleThick	A thick style divider is displayed between subviews. This is the default.
NSSplitViewDividerStyleThin	A thin style divider is displayed between subviews.

NSStringDrawing.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSStringDrawingTruncatesLastVisibleLine	Truncates and adds the ellipsis character to the last visible line if the text doesn't fit into the bounds specified.

NSTableView.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSTableViewSelectionHighlightStyle	The following constants specify the selection highlight styles. These constants are used by selectionHighlightStyle and setSelectionHighlightStyle:.
NSTableViewSelectionHighlightStyleRegular	The regular highlight style of NSTableView.
NSTableViewSelectionHighlightStyleSourceList	The source list style of NSTableView.

NSTextView.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAllRomanInputSourcesLocaleIdentifier	A meta-locale identifier representing the set of Roman input sources available. You can pass [NSArray arrayWithObject: NSAllRomanInputSources-LocaleIdentifier] to the setAllowedInputSourceLocales: method to restrict allowed input sources to Roman only.
NSFindPanelCaseInsensitiveSearch	Boolean value specifying whether the search is case-insensitive. YES specifies a case-insensitive search; otherwise, NO.
NSFindPanelSearchOptionsPboardType	Type for NSFindPanel metadata property list. Used with the NSPasteBoard method propertyListForType:.
NSFindPanelSubstringMatch	NSNumber object containing one of the values defined in "NSFindPanelSubstringMatchType."
NSFindPanelSubstringMatchType	The type of substring matching used by the Find panel.

NSFindPanelSubstringMatchTypeContains	Finds a word containing the search string.
NSFindPanelSubstringMatchTypeEndsWith	Finds a word ending with the search string.
NSFindPanelSubstringMatchTypeFullWord	Finds a word exactly matching the search string.
NSFindPanelSubstringMatchTypeStartsWith	Finds a word starting with the search string.

NSTrackingArea.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSTrackingActiveAlways	The owner receives messages regardless of first-responder status, window status, or application status. The cursorUpdate: message is not sent when the NSTrackingCursorUpdate option is specified along with this constant.
NSTrackingActiveInActiveApp	The owner receives messages when the application is active.
NSTrackingActiveInKeyWindow	The owner receives messages when the view in in the key window.
NSTrackingActiveWhenFirstResponder	The owner receives messages when the view is the first responder.
NSTrackingAreaOptions	
NSTrackingAssumeInside	The first event is generated when the cursor leaves the tracking area, regardless if the cursor is inside the area when the NSTrackingArea is added to a view. If this option is not specified, the first event is generated when the cursor leaves the tracking area if the cursor is initially inside the area, or when the cursor enters the area if the cursor is initially outside it. Generally, you do not want to request this behavior.
NSTrackingCursorUpdate	The owner of the tracking area receives cursorUpdate: messages when the mouse cursor enters the area; when the mouse leaves the area, the cursor is appropriately reset.
NSTrackingEnabledDuringMouseDrag	The owner receives NSMouseEntered events when the mouse cursor is dragged into the tracking area. If this option is not specified, the owner receives mouse-entered events when the mouse is moved (no buttons pressed) into the tracking area and on NSLeftMouseUp events after a mouse drag.

NSTrackingInVisibleRect	Mouse tracking occurs only in the visible rectangle of the view—in other words, that region of the tracking rectangle that is unobscured. Otherwise, the entire tracking area is active regardless of overlapping views. The NSTrackingArea object is automatically synchronized with changes in the view's visible area (visibleRect) and the value returned from rect is ignored.
NSTrackingMouseEnteredAndExited	The owner of the tracking area receives mouseEntered: when the mouse cursor enters the area and mouseExited: events when the mouse leaves the area.
NSTrackingMouseMoved	The owner of the tracking area receives mouseMoved: messages while the mouse cursor is within the area.

NSView.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSFullScreenModeAllScreens	Key whose corresponding value specifies whether the view should take over all screens.
NSFullScreenModeSetting	Key whose corresponding value specifies the the full screen mode setting.
NSFullScreenModeWindowLevel	Key whose corresponding value specifies the screen mode window level.
NSViewDidUpdateTrackingAreasNotification	Posted whenever an NSView object recalculates its tracking areas.

NSWindow.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSWindowBackingLocation	These constants and data type represent a window's possible backing locations.
NSWindowBackingLocationDefault	Determined by the operating system.
NSWindowBackingLocationMainMemory	Physical memory.

NSWindowBackingLocationVideoMemory	Video memory.
NSWindowCollectionBehavior	These constants and data type identify window behavior in relation to window browsers and organizers, such as Spaces.
NSWindowCollectionBehaviorCanJoinAllSpaces	The window appears in all spaces.
NSWindowCollectionBehaviorDefault	The window can be associated to one space at a time.
NSWindowCollectionBehaviorMoveToActiveSpace	Making the window active does not cause a space switch; the window switches to the active space.
NSWindowSharingNone	The window's contents cannot be read by another process.
NSWindowSharingReadOnly	The window's contents can be read but not modified by another process.
NSWindowSharingReadWrite	The window's contents can be read and modified by another process.
NSWindowSharingType	These constants and data type represent the access levels other processes can have to a window's content.

10.4 Symbol Changes

This article lists the symbols added to `AppKit.framework` in Mac OS X v10.4.

Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

CIColor

Complete reference information is available in the `CIColor` reference.

Instance Methods

`initWithColor:`	Initializes a newly allocated CIColor object using an NSColor object.

CIImage

Complete reference information is available in the `CIImage` reference.

Instance Methods

`drawAtPoint:fromRect:operation:fraction:`	Draws all or part of the image at the specified point in the current coordinate system.
`drawInRect:fromRect:operation:fraction:`	Draws all or part of the image in the specified rectangle in the current coordinate system
`initWithBitmapImageRep:`	Initializes the receiver, a newly allocated CIImage object, with the specified bitmap.

NSActionCell

Complete reference information is available in the `NSActionCell` reference.

Instance Methods

`setControlView:`	Sets the receiver's control view, the view in which it is drawn.

NSAlert

Complete reference information is available in the `NSAlert` reference.

Class Methods

`alertWithError:`	Returns an alert initialized from information in an error object.

NSAnimation (New)

Complete reference information is available in the `NSAnimation` reference.

Instance Methods

`addProgressMark:`	Adds the progress mark to the receiver.
`animationBlockingMode`	Returns the blocking mode the receiver is next scheduled to run under.
`animationCurve`	Returns the animation curve the receiver is running under.
`clearStartAnimation`	Clears linkage to another animation that causes the receiver to start.
`clearStopAnimation`	Clears linkage to another animation that causes the receiver to stop.
`currentProgress`	Returns the current progress of the receiver.
`currentValue`	Returns the current value of the effect based on the current progress.
`delegate`	Returns the delegate of the receiver.
`duration`	Returns the duration of the animation, in seconds.
`frameRate`	Returns the frame rate of the animation.
`initWithDuration:animationCurve:`	Returns an NSAnimation object initialized with the specified duration and animation-curve values.
`isAnimating`	Returns a Boolean value that indicates whether the receiver is currently animating.

`progressMarks`	Returns the receiver’s progress marks.
`removeProgressMark:`	Removes progress mark from the receiver.
`runLoopModesForAnimating`	Overridden to return the run-loop modes that the receiver uses to run the animation timer in.
`setAnimationBlockingMode:`	Sets the blocking mode of the receiver.
`setAnimationCurve:`	Sets the receiver’s animation curve.
`setCurrentProgress:`	Sets the current progress of the receiver.
`setDelegate:`	Sets the delegate of the receiver.
`setDuration:`	Sets the duration of the animation to a specified number of seconds.
`setFrameRate:`	Sets the frame rate of the receiver.
`setProgressMarks:`	Sets the receiver’s progress marks to the values specified in the passed-in array.
`startAnimation`	Starts the animation represented by the receiver.
`startWhenAnimation:reachesProgress:`	Starts running the animation represented by the receiver when another animation reaches a specific progress mark.
`stopAnimation`	Stops the animation represented by the receiver.
`stopWhenAnimation:reachesProgress:`	Stops running the animation represented by the receiver when another animation reaches a specific progress mark.

Delegate Methods

`animation:didReachProgressMark:`	Sent to the delegate when an animation reaches a specific progress mark.
`animation:valueForProgress:`	Requests a custom curve value for the current progress value.
`animationDidEnd:`	Sent to the delegate when the specified animation completes its run.
`animationDidStop:`	Sent to the delegate when the specified animation is stopped before it completes its run.
`animationShouldStart:`	Sent to the delegate just after an animation is started.

NSApplication

Complete reference information is available in the `NSApplication` reference.

Delegate Methods

`application:printFiles:withSettings:`
`showPrintPanels:`
	Prints a group of files.

`application:willPresentError:`	Sent to the delegate before the specified application presents an error message to the user.

NSArrayController

Complete reference information is available in the `NSArrayController` reference.

Instance Methods

`add:`	Creates and adds a new object to the receiver's content and arranged objects.
`alwaysUsesMultipleValuesMarker`	Returns a Boolean value that indicates whether the receiver always returns the multiple values marker when multiple objects are selected.
`clearsFilterPredicateOnInsertion`	Returns a Boolean value that indicates whether the receiver automatically clears an existing filter predicate when new items are inserted or added to the content.
`filterPredicate`	Returns the predicate used by the receiver to filter the array controller contents.
`remove:`	Removes the receiver's selected objects from the content collection.
`setAlwaysUsesMultipleValuesMarker:`	Sets whether the receiver always returns the multiple values marker when multiple objects are selected.
`setClearsFilterPredicateOnInsertion:`	Sets whether the receiver automatically clears an existing filter predicate when a new object is inserted or added to the content array.
`setFilterPredicate:`	Sets the predicate used to filter the contents of the receiver.

NSAttributedString

Complete reference information is available in the `NSAttributedString` reference.

Instance Methods

boundingRectWithSize:options:	Calculates and returns bounding rectangle for the receiver drawn using the options specified, within the given rectangle in the current graphics context.
dataFromRange:documentAttributes:error:	Returns an NSData object that contains a text stream corresponding to the characters and attributes within the given range.
drawWithRect:options:	Draws the receiver with the specified options, within the given rectangle in the current graphics context.
fileWrapperFromRange:documentAttributes:error:	Returns an NSFileWrapper object that contains a text stream corresponding to the characters and attributes within the given range.
initWithData:options:documentAttributes:error:	Initializes and returns a new NSAttributedString object from the data contained in the given NSData object.
initWithURL:options:documentAttributes:error:	Initializes a new NSAttributedString object from the contents of the given URL.
itemNumberInTextList:atIndex:	Returns the range of the item at the given index within the given list.
rangeOfTextBlock:atIndex:	Returns the range of the individual text block that contains the given location.
rangeOfTextList:atIndex:	Returns the range of the given text list that contains the given location.
rangeOfTextTable:atIndex:	Returns the range of the given text table that contains the given location

NSBitmapImageRep

Complete reference information is available in the NSBitmapImageRep reference.

Instance Methods

bitmapFormat	Returns the bitmap format of the receiver.
colorAtX:y:	Returns the color of the pixel at the specified coordinates.

getPixel:atX:y:	Returns by indirection the pixel data for the specified location in the receiver.
initWithBitmapDataPlanes:pixelsWide:pixelsHigh: bitsPerSample:samplesPerPixel:hasAlpha:isPlanar: colorSpaceName:bitmapFormat:bytesPerRow: bitsPerPixel:	Initializes the receiver, a newly allocated NSBitmapImageRep object, so it can render the specified image.
setColor:atX:y:	Changes the color of the pixel at the specified coordinates.
setPixel:atX:y:	Sets the receiver's pixel at the specified coordinates to the specified raw pixel values.

NSButtonCell

Complete reference information is available in the `NSButtonCell` reference.

Instance Methods

backgroundColor	Returns the background color of the receiver.
drawBezelWithFrame:inView:	Draws the border of the button using the current bezel style.
drawImage:withFrame:inView:	Draws the image associated with the button’s current state.
drawTitle:withFrame:inView:	Draws the button’s title centered vertically in a specified rectangle.
setBackgroundColor:	Sets the background color of the receiver.

NSCell

Complete reference information is available in the `NSCell` reference.

Instance Methods

allowsUndo	Returns a Boolean value that indicates whether the receiver assumes responsibility for undo operations.
baseWritingDirection	Returns the initial writing direction used to determine the actual writing direction for text.
lineBreakMode	Returns the line break mode currently used when drawing text.
setAllowsUndo:	Sets whether the receiver assumes responsibility for undo operations within the cell.

setBaseWritingDirection:	Sets the initial writing direction used to determine the actual writing direction for text .
setControlView:	Sets the receiver’s control view.
setLineBreakMode:	Sets the line break mode to use when drawing text

NSCIImageRep (New)

Complete reference information is available in the NSCIImageRep reference.

Class Methods

imageRepWithCIImage:	Creates and returns an NSCIImageRep object initialized to the specified CIImage instance.

Instance Methods

CIImage	Returns the receiver's CIImage instance.
initWithCIImage:	Returns an NSCIImageRep object initialized to the specified CIImage instance.

NSColor

Complete reference information is available in the NSColor reference.

Class Methods

colorWithCIColor:	Converts a Core Image color object to its NSColor equivalent.
colorWithColorSpace:components:count:	Returns an NSColor object created from the specified components of the given color space.

Instance Methods

colorSpace	Returns an object representing the color space of the receiver.
colorUsingColorSpace:	Returns a new color object representing the color of the receiver in the specified color space.
getComponents:	Returns the components of the receiver as an array.

`numberOfComponents`	Returns the number of components in the receiver.

NSColorSpace (New)

Complete reference information is available in the `NSColorSpace` reference.

Class Methods

`deviceCMYKColorSpace`	Returns an NSColorSpace object representing a calibrated or device-dependent CMYK color space.
`deviceGrayColorSpace`	Returns an NSColorSpace object representing a calibrated or device-dependent gray color space.
`deviceRGBColorSpace`	Returns an NSColorSpace object representing a calibrated or device-dependent RGB color space.
`genericCMYKColorSpace`	Returns an NSColorSpace object representing a device-independent CMYK color space.
`genericGrayColorSpace`	Returns an NSColorSpace object representing a device-independent gray color space.
`genericRGBColorSpace`	Returns an NSColorSpace object representing a device-independent RGB color space.

Instance Methods

`colorSpaceModel`	Returns the model on which the color space of the receiver is based.
`colorSyncProfile`	Returns the ColorSync profile from which the receiver was created.
`ICCProfileData`	Returns the ICC profile data from which the receiver was created.
`initWithColorSyncProfile:`	Initializes and returns an NSColorSpace object given a ColorSync profile.
`initWithICCProfileData:`	Initializes and returns an NSColorSpace object given an ICC profile.
`localizedName`	Returns the localized name of the receiver.
`numberOfColorComponents`	Returns the number of components supported by the receiver.

NSControl

Complete reference information is available in the `NSControl` reference.

Instance Methods

`baseWritingDirection`	Returns the initial writing direction used to determine the actual writing direction for text.
`setBaseWritingDirection:`	Sets the initial writing direction used to determine the actual writing direction for text .

NSController

Complete reference information is available in the `NSController` reference.

Instance Methods

`commitEditingWithDelegate:didCommitSelector:contextInfo:`	Attempts to commit any pending changes in known editors of the receiver.

NSDatePicker (New)

Complete reference information is available in the `NSDatePicker` reference.

Instance Methods

`backgroundColor`	Returns the background color of the receiver.
`calendar`	Returns the calendar used by the receiver.
`datePickerElements`	Returns a bitmask that indicates which visual elements of the date picker are currently shown, and which won't be usable because they are hidden.
`datePickerMode`	Returns the receiver’s date picker mode.
`datePickerStyle`	Returns the receiver’s date picker style.
`dateValue`	Returns the receiver’s date.
`delegate`	Returns the delegate of the receiver’s date picker cell.
`drawsBackground`	Returns whether the receiver draws the background.
`isBezeled`	Returns whether the receiver has a bezeled border.
`isBordered`	Returns whether the receiver has a plain border.
`locale`	Returns the receiver’s locale.
`maxDate`	Returns the maximum date value the receiver allows as input.

`minDate`	Returns the minimum date value the receiver allows as input.
`setBackgroundColor:`	Sets the receiver's background color.
`setBezeled:`	Specifies whether the receiver draws a bezeled border.
`setBordered:`	Specifies whether the receiver draws a plain border.
`setCalendar:`	Sets the receiver's calendar.
`setDatePickerElements:`	Sets a bitmask that indicates which visual elements of the date picker are currently shown, and which won't be usable because they are hidden.
`setDatePickerMode:`	Sets the receiver's date picker mode.
`setDatePickerStyle:`	Sets the receiver's date picker style.
`setDateValue:`	Sets the receiver's date to a new starting value.
`setDelegate:`	Sets the delegate of the receiver's date picker cell.
`setDrawsBackground:`	Specifies whether the receiver draws the background.
`setLocale:`	Sets the receiver's locale.
`setMaxDate:`	Sets the maximum date allowed as input by the receiver.
`setMinDate:`	Sets the minimum date allowed as input by the receiver.
`setTextColor:`	Sets the text color of the receiver.
`setTimeInterval:`	Sets the time interval of the date range.
`setTimeZone:`	Sets the receiver's time zone.
`textColor`	Returns the text color of the receiver.
`timeInterval`	Returns the time interval that represents the date range.
`timeZone`	Returns the receiver's time zone.

NSDatePickerCell (New)

Complete reference information is available in the `NSDatePickerCell` reference.

Instance Methods

`backgroundColor`	Returns the receiver's background color.
`calendar`	Returns the calendar used by the receiver.

datePickerElements	Returns a bitmask that indicates which visual elements of the date picker are currently shown, and which won't be usable because they are hidden.
datePickerMode	Returns the receiver’s date picker mode.
datePickerStyle	Returns the receiver’s date picker style.
dateValue	Returns the receiver’s date.
delegate	Returns the receiver’s delegate.
drawsBackground	Returns whether the receiver draws the background.
locale	Returns the receiver’s locale.
maxDate	Returns the maximum date value that the receiver allows as input.
minDate	Returns the minimum date value that the receiver allows as input.
setBackgroundColor:	Sets the receiver’s background color
setCalendar:	Sets the receiver’s calendar.
setDatePickerElements:	Sets a bitmask that indicates which visual elements of the date picker are currently shown, and which won't be usable because they are hidden.
setDatePickerMode:	Sets the receiver’s date picker mode.
setDatePickerStyle:	Sets the receiver’s date picker style.
setDateValue:	Sets the receiver’s date to a new starting value.
setDelegate:	Sets the receiver's delegate.
setDrawsBackground:	Sets whether the receiver draws the background.
setLocale:	Sets the receiver’s locale.
setMaxDate:	Sets the maximum date allowed as input by the receiver to the given date.
setMinDate:	Sets the minimum date allowed as input by the receiver to the given date.
setTextColor:	Sets the receiver’s text color.
setTimeInterval:	Sets the time interval of the date range.
setTimeZone:	Sets the receiver’s time zone.
textColor	Returns the receiver’s text color.
timeInterval	Returns the time interval that represents the date range.

`timeZone`	Returns the receiver's time zone.

Delegate Methods

`datePickerCell:validateProposedDateValue:timeInterval:`	The delegate receives this message each time the user attempts to change the receiver's value, allowing the delegate the opportunity to override the change.

NSDocument

Complete reference information is available in the `NSDocument` reference.

Instance Methods

`autosavedContentsFileURL`	Returns the location of the most recently autosaved document contents.
`autosaveDocumentWithDelegate:didAutosaveSelector:contextInfo:`	Autosaves the document's contents at an appropriate location.
`autosavingFileType`	Returns the document type that should be used for an autosave operation.
`dataOfType:error:`	Creates and returns a data object that contains the contents of the document, formatted to a specified type.
`fileAttributesToWriteToURL:ofType:forSaveOperation:originalContentsURL:error:`	As a file is being saved, returns the attributes that should be written to a file or file package located by a URL, formatted to a specified type, for a particular kind of save operation.
`fileModificationDate`	Returns the last known modification date of the document's on-disk representation.
`fileURL`	Returns the location of the document's on-disk representation.
`fileWrapperOfType:error:`	Creates and returns a file wrapper that contains the contents of the document, formatted to the specified type.
`hasUnautosavedChanges`	Return YES if the document has changes that have not been autosaved, as determined by the history of previous invocations of updateChangeCount:.

`initForURL:withContentsOfURL:ofType:error:`	Initializes a document located by a URL of a specified type, but by reading the contents for the document from a different URL.
`initWithContentsOfURL:ofType:error:`	Initializes a document located by a URL of a specified type.
`initWithType:error:`	Initializes a document of a specified type.
`presentError:`	Presents an error alert to the user as a modal panel.

`presentError:modalForWindow:delegate:`
`didPresentSelector:contextInfo:`
	Presents an error alert to the user as a modal panel.

`printDocumentWithSettings:showPrintPanel:`
`delegate:didPrintSelector:contextInfo:`
	Prints the document.

`printOperationWithSettings:error:`	Creates a print operation and returns it if successful.
`readFromData:ofType:error:`	Sets the contents of this document by reading from data of a specified type and returns YES if successful.
`readFromFileWrapper:ofType:error:`	Sets the contents of this document by reading from a file wrapper of a specified type.
`readFromURL:ofType:error:`	Sets the contents of this document by reading from a file or file package, of a specified type, located by a URL.
`revertToContentsOfURL:ofType:error:`	Discards all unsaved document modifications and replaces the document's contents by reading a file or file package located by a URL of a specified type.

`saveToURL:ofType:forSaveOperation:delegate:`
`didSaveSelector:contextInfo:`
	Saves the contents of the document to a file or file package located by a URL, formatted to a specified type, for a particular kind of save operation.

`saveToURL:ofType:forSaveOperation:error:`	Saves the contents of the document to a file or file package located by a URL, formatted to a specified type, for a particular kind of save operation, and returns YES if successful.
`setAutosavedContentsFileURL:`	Sets the location of the most recently autosaved document contents.

`setFileModificationDate:`	Sets the last known modification date of the document's on-disk representation to the given modification date.
`setFileURL:`	Sets the location of the document's on-disk representation.
`willPresentError:`	Called when the receiver is about to present an error.
`writableTypesForSaveOperation:`	Returns the names of the types to which this document can be saved for a specified kind of save operation.
`writeSafelyToURL:ofType:forSaveOperation:error:`	Writes the contents of the document to a file or file package located by a URL.
`writeToURL:ofType:error:`	Writes the contents of the document to a file or file package located by a URL, formatted to a specified type.
`writeToURL:ofType:forSaveOperation:originalContentsURL:error:`	Writes the contents of the document to a file or file package located by a URL.

NSDocumentController

Complete reference information is available in the `NSDocumentController` reference.

Instance Methods

`autosavingDelay`	Returns the time interval in seconds for periodic autosaving.
`defaultType`	Returns the name of the document type that should be used when creating new documents.
`documentClassNames`	Returns the names of NSDocument subclasses supported by this application.
`documentForURL:`	Returns, for a given URL, the open document whose file or file package is located by the URL, or nil if there is no such open document.

makeDocumentForURL:withContentsOfURL:ofType:error:	Instantiates a document located by a URL, of a specified type, but by reading the contents for the document from another URL, and returns it if successful.
makeDocumentWithContentsOfURL:ofType:error:	Instantiates a document located by a URL, of a specified type, and returns it if successful.
makeUntitledDocumentOfType:error:	Instantiates a new untitled document of the specified type and returns it if successful.
maximumRecentDocumentCount	Returns the maximum number of items that may be presented in the standard Open Recent menu.
openDocumentWithContentsOfURL:display:error:	Opens a document located by the given URL presents its user interface if requested, and returns the document if successful.
openUntitledDocumentAndDisplay:error:	Creates a new untitled document, presents its user interface if displayDocument is YES, and returns the document if successful.
presentError:	Presents an error alert to the user as a modal panel.
presentError:modalForWindow:delegate: didPresentSelector:contextInfo:	Presents an error alert to the user as a modal panel.
reopenDocumentForURL:withContentsOfURL:error:	Reopens an autosaved document located by a URL, by reading the contents for the document from another URL, presents its user interface, and returns YES if successful.
setAutosavingDelay:	Sets the time interval in seconds for periodic autosaving.
typeForContentsOfURL:error:	Returns, for a specified URL, the name of the document type that should be used when opening the document at that location, if successful.
willPresentError:	Called when the receiver is about to present an error, returns the error that should actually be presented.

NSEvent

Complete reference information is available in the NSEvent reference.

Instance Methods

absoluteX	Reports the absolute x coordinate of a pointing device on its tablet at full tablet resolution.
absoluteY	Reports the absolute y coordinate of a pointing device on its tablet at full tablet resolution.
absoluteZ	Reports the absolute z coordinate of pointing device on its tablet at full tablet resolution.
buttonMask	Returns a bit mask identifying the buttons pressed when the tablet event represented by the receiver was generated.
capabilityMask	Returns a mask whose set bits indicate the capabilities of the tablet device that generated the event represented by the receiver.
deviceID	Returns a special identifier that is used to match tablet-pointer events with the tablet-proximity event represented by the receiver.
isEnteringProximity	Returns YES to indicate that a pointing device is entering the proximity of its tablet and NO when it is leaving it.
pointingDeviceID	Returns the index of the pointing device currently in proximity with the tablet.
pointingDeviceSerialNumber	Returns the vendor-assigned serial number of a pointing device of a certain type.
pointingDeviceType	Returns a NSPointingDeviceType constant indicating the kind of pointing device associated with the receiver.
rotation	Returns the rotation in degrees of the tablet pointing device associated with the receiver.
systemTabletID	Returns the index of the tablet device connected to the system.
tabletID	Returns the USB model identifier of the tablet device associated with the receiver.
tangentialPressure	Reports the tangential pressure on the device that generated the event represented by the receiver.
tilt	Reports the scaled tilt values of the pointing device that generated the event represented by the receiver.
uniqueID	Returns the unique identifier of the pointing device that generated the event represented by the receiver.

vendorDefined	Returns an array of three vendor-defined NSNumber objects associated with the pointing-type event represented by the receiver.
vendorID	Returns the vendor identifier of the tablet associated with the receiver.
vendorPointingDeviceType	Returns a coded bit field whose set bits indicate the type of pointing device (within a vendor selection) associated with the receiver.

NSFont

Complete reference information is available in the NSFont reference.

Class Methods

fontWithDescriptor:size:	Returns a font object for the specified font descriptor and font size.
fontWithDescriptor:textTransform:	Returns a font object for the specified font descriptor and text transform.

Instance Methods

getAdvancements:forGlyphs:count:	Returns an array of the advancements for the specified glyphs rendered by the receiver.
getAdvancements:forPackedGlyphs:length:	Returns an array of the advancements for the specified packed glyphs and rendered by the receiver.
getBoundingRects:forGlyphs:count:	Returns an array of the bounding rectangles for the specified glyphs rendered by the receiver.
leading	Returns the receiver's leading.
renderingMode	Returns the rendering mode of the receiver.
screenFontWithRenderingMode:	Returns a bitmapped screen font, when sent to a font object representing a scalable PostScript font, with the specified rendering mode, matching the receiver in typeface and matrix (or size), or nil if such a font can't be found.
setInContext:	Establishes the receiver as the current font for the specified graphics context.
textTransform	Returns the current transformation matrix for the receiver.

NSFontDescriptor

Complete reference information is available in the `NSFontDescriptor` reference.

Class Methods

`fontDescriptorWithName:matrix:`	Returns a font descriptor with the NSFontNameAttribute and NSFontMatrixAttribute dictionary attributes set to the given values.

Instance Methods

`fontDescriptorByAddingAttributes:`	Returns a new font descriptor that is the same as the receiver but with the specified attributes taking precedence over the existing ones.
`fontDescriptorWithFace:`	Returns a new font descriptor that is the same as the receiver but with the specified face.
`fontDescriptorWithFamily:`	Returns a new font descriptor whose attributes are the same as the receiver but from the specified family.
`fontDescriptorWithMatrix:`	Returns a new font descriptor that is the same as the receiver but with the specified matrix.
`fontDescriptorWithSize:`	Returns a new font descriptor that is the same as the receiver but with the specified point size.
`fontDescriptorWithSymbolicTraits:`	Returns a new font descriptor that is the same as the receiver but with the specified symbolic traits taking precedence over the existing ones.
`matchingFontDescriptorsWithMandatoryKeys:`	Returns all the fonts available on the system whose specified attributes match those of the receiver.
`matrix`	Returns the current transform matrix of the receiver.
`objectForKey:`	Returns the font attribute specified by the given key.
`pointSize`	Returns the point size of the receiver.
`postscriptName`	Returns the PostScript name of the receiver.
`symbolicTraits`	Returns a bit mask that describes the traits of the receiver.

NSForm

Complete reference information is available in the `NSForm` reference.

Instance Methods

`setTextBaseWritingDirection:`	Sets the writing direction for the text content of every control embedded in the form.
`setTitleBaseWritingDirection:`	Sets the writing direction for the title of every control embedded in the form.

NSFormCell

Complete reference information is available in the `NSFormCell` reference.

Instance Methods

`placeholderAttributedString`	Returns the cell's attributed placeholder string.
`placeholderString`	Returns the cell's plain text placeholder string.
`setPlaceholderAttributedString:`	Sets the attributed placeholder string for the cell.
`setPlaceholderString:`	Sets the plain-text placeholder string for the cell.
`setTitleBaseWritingDirection:`	Sets the default writing direction used to render the form cell's title.
`titleBaseWritingDirection`	Returns the default writing direction used to render the form cell's title.

NSGraphicsContext

Complete reference information is available in the `NSGraphicsContext` reference.

Class Methods

`graphicsContextWithBitmapImageRep:`	Instantiates and returns a new graphics context using the supplied NSBitmapImageRep object as the context destination.
`graphicsContextWithGraphicsPort:flipped:`	Instantiates and returns a new graphics context from the given graphics port.

Instance Methods

`CIContext`	Returns a CIContext object that you can use to render into the receiver.
`compositingOperation`	Returns the receiver's global compositing operation setting.
`isFlipped`	Returns a Boolean value that indicates the receiver's flipped state.
`setCompositingOperation:`	Sets the receiver's global compositing operation.

NSImageView

Complete reference information is available in the `NSImageView` reference.

Instance Methods

`allowsCutCopyPaste`	Returns a Boolean value indicating whether the receiver allows the user to cut, copy and paste of the image contents.
`setAllowsCutCopyPaste:`	Sets whether the receiver allows the user to cut, copy and paste the image contents.

NSLayoutManager

Complete reference information is available in the `NSLayoutManager` reference.

Instance Methods

`boundsRectForTextBlock:atIndex:effectiveRange:`	Returns the bounding rectangle within which the given text block containing the glyph at the given index is to be laid out.
`boundsRectForTextBlock:glyphRange:`	Returns the bounding rectangle enclosing the given text block containing the given glyph range.
`glyphGenerator`	Returns the glyph generator used by this layout manager.
`layoutRectForTextBlock:atIndex:effectiveRange:`	Returns the layout rectangle within which the given text block containing the glyph at the given index is to be laid out.
`layoutRectForTextBlock:glyphRange:`	Returns the layout rectangle within which the given text block containing the given glyph range is to be laid out.

lineFragmentRectForGlyphAtIndex:effectiveRange:withoutAdditionalLayout:	Returns the line fragment rectangle containing the glyph at the given glyph index.
lineFragmentUsedRectForGlyphAtIndex:effectiveRange:withoutAdditionalLayout:	Returns the usage rectangle for the line fragment in which the given glyph is laid and (optionally) by reference the whole range of glyphs that are in that fragment.
setBoundsRect:forTextBlock:glyphRange:	Sets the bounding rectangle enclosing a given text block containing the given glyph range.
setGlyphGenerator:	Sets the glyph generator used by this layout manager.
setLayoutRect:forTextBlock:glyphRange:	Sets the layout rectangle enclosing the given text block containing the given glyph range.
textContainerForGlyphAtIndex:effectiveRange:withoutAdditionalLayout:	Returns the container in which the given glyph is laid out and (optionally) by reference the whole range of glyphs that are in that container.

NSLevelIndicator (New)

Complete reference information is available in the NSLevelIndicator reference.

Instance Methods

criticalValue	Returns the receiver's critical value.
maxValue	Returns the receiver's maximum value.
minValue	Returns the receiver's minimum value.
numberOfMajorTickMarks	Returns the number of major tick marks associated with the receiver.
numberOfTickMarks	Returns the number of tick marks associated with the receiver.
rectOfTickMarkAtIndex:	Returns the bounding rectangle of the tick mark identified by index (the minimum-value tick mark is at index 0).
setCriticalValue:	Sets the receiver's critical value to criticalValue.
setMaxValue:	Sets the maximum value the receiver can represent to maxValue.
setMinValue:	Sets the minimum value the receiver can represent to minValue.
setNumberOfMajorTickMarks:	Sets the number of major tick marks displayed by the receiver.

`setNumberOfTickMarks:`	Sets the number of tick marks displayed by the receiver (which include those assigned to the minimum and maximum values) to count.
`setTickMarkPosition:`	Sets where tick marks appear relative to the receiver.
`setWarningValue:`	Sets the receiver's warning value to warningValue.
`tickMarkPosition`	Returns how the receiver's tick marks are aligned with it.
`tickMarkValueAtIndex:`	Returns the receiver's value represented by the tick mark at index (the minimum-value tick mark has an index of 0).
`warningValue`	Returns the receiver's warning value.

NSLevelIndicatorCell (New)

Complete reference information is available in the `NSLevelIndicatorCell` reference.

Instance Methods

`criticalValue`	Returns the receiver's critical value.
`initWithLevelIndicatorStyle:`	Initializes the receiver with the style specified by levelIndicatorStyle.
`levelIndicatorStyle`	Returns the level indicator style of the receiver.
`maxValue`	Returns the receiver's maximum value.
`minValue`	Returns the receiver's minimum value.
`numberOfMajorTickMarks`	Returns the number of major tick marks associated with the receiver.
`numberOfTickMarks`	Returns the number of tick marks associated with the receiver.
`rectOfTickMarkAtIndex:`	Returns the bounding rectangle of the tick mark identified by index (the minimum-value tick mark is at index 0).
`setCriticalValue:`	Sets the receiver's critical value to criticalValue.
`setImage:`	Sets the image displayed by the receiver for the NSRatingLevelIndicatorStyle to image.
`setLevelIndicatorStyle:`	Sets the style of the receiver to levelIndicatorStyle.
`setMaxValue:`	Sets the maximum value the receiver can represent to maxValue.
`setMinValue:`	Sets the minimum value the receiver can represent to minValue.
`setNumberOfMajorTickMarks:`	Sets the number of major tick marks displayed by the receiver.

setNumberOfTickMarks:	Sets the number of tick marks displayed by the receiver (which include those assigned to the minimum and maximum values) to numberOfTickMarks.
setTickMarkPosition:	Sets where tick marks appear relative to the receiver.
setWarningValue:	Sets the receiver's warning value to warningValue.
tickMarkPosition	Returns how the receiver's tick marks are aligned with it.
tickMarkValueAtIndex:	Returns the receiver's value represented by the tick mark at index (the minimum-value tick mark has an index of 0).
warningValue	Returns the receiver's warning value.

NSMenu

Complete reference information is available in the `NSMenu` reference.

Instance Methods

menuBarHeight	Returns the menu bar height for the current application's main menu.

NSMenuItemCell

Complete reference information is available in the `NSMenuItemCell` reference.

Instance Methods

tag	Returns the integer tag of the selected menu item, or 0 if no item is selected.

NSMutableAttributedString

Complete reference information is available in the `NSMutableAttributedString` reference.

Instance Methods

readFromData:options:documentAttributes:error:	Sets the contents of the receiver from the stream at data.
readFromURL:options:documentAttributes:error:	Sets the contents of receiver from the file at url.

setBaseWritingDirection:range:	Sets the base writing direction for the characters in range to writingDirection.

NSMutableParagraphStyle

Complete reference information is available in the NSMutableParagraphStyle reference.

Instance Methods

setHeaderLevel:	Specifies whether the paragraph is to be treated as a header for purposes of HTML generation.
setHyphenationFactor:	Specifies the threshold for hyphenation.
setTextBlocks:	Sets the text blocks containing the paragraph, nested from outermost to innermost to array.
setTextLists:	Sets the text lists containing the paragraph, nested from outermost to innermost, to array.
setTighteningFactorForTruncation:	Specifies the threshold for using tightening as an alternative to truncation.

NSObject

Complete reference information is available in the NSObject reference.

Instance Methods

accessibilitySetOverrideValue:forAttribute:	Overrides the receiver's attribute attribute, or adds it if it does not exist, and sets its value to value.
commitEditingWithDelegate:didCommitSelector:contextInfo:	Attempt to commit any currently edited results of the receiver.
infoForBinding:	Returns a dictionary describing the receiver's binding.
wantsPeriodicDraggingUpdates	Asks the destination object whether it wants to receive periodic draggingUpdated: messages.
windowDidChangeScreenProfile:	Sent by the default notification center immediately after an NSWindow object has changed screen display profiles.

NSObjectController

Complete reference information is available in the `NSObjectController` reference.

Instance Methods

`entityName`	Returns the entity name used by the receiver to create new objects.
`fetch:`	Causes the receiver to fetch the data objects specified by the entity name and fetch predicate.
`fetchPredicate`	Returns the receiver's fetch predicate.
`fetchWithRequest:merge:error:`	Subclasses should override this method to customize a fetch request, for example to specify fetch limits.
`managedObjectContext`	Returns the receiver's managed object context.
`setEntityName:`	Sets the entity name used by the receiver to create new objects.
`setFetchPredicate:`	Sets the receiver's fetch predicate.
`setManagedObjectContext:`	Sets the receiver's managed object context.

NSOutlineView

Complete reference information is available in the `NSOutlineView` reference.

Delegate Methods

`outlineView:heightOfRowByItem:`	Returns the height in points of the row containing item.
`outlineView:namesOfPromisedFilesDroppedAtDestination:forDraggedItems:`	Returns an array of filenames for the created files that the receiver promises to create.
`outlineView:toolTipForCell:rect:tableColumn:item:mouseLocation:`	When the cursor pauses over a given cell, the value returned from this method is displayed in a tooltip.

NSParagraphStyle

Complete reference information is available in the `NSParagraphStyle` reference.

Instance Methods

headerLevel	Specifies whether the paragraph is to be treated as a header for purposes of HTML generation.
hyphenationFactor	Returns the paragraph's threshold for hyphenation.
textBlocks	Returns an array specifying the text blocks containing the paragraph.
textLists	Returns an array specifying the text lists containing the paragraph.
tighteningFactorForTruncation	Returns the threshold for using tightening as an alternative to truncation.

NSPersistentDocument (New)

Complete reference information is available in the NSPersistentDocument reference.

Instance Methods

configurePersistentStoreCoordinatorForURL:ofType:error:	Configures the receiver's persistent store coordinator for a given URL and document type.
managedObjectContext	Returns the managed object context for the receiver.
managedObjectModel	Returns the receiver's managed object model.
persistentStoreTypeForFileType:	Returns the type of persistent store associated with the specified file type.
readFromURL:ofType:error:	Sets the contents of the receiver by reading from a file of a given type located by a given URL.
revertToContentsOfURL:ofType:error:	Overridden to clean up the managed object context and controllers during a revert.
setManagedObjectContext:	Sets the receiver's managed object context.
writeToURL:ofType:forSaveOperation:originalContentsURL:error:	Saves changes in the document's managed object context and saves the document's persistent store to a given URL.

NSPopUpButton

Complete reference information is available in the `NSPopUpButton` reference.

Instance Methods

`selectItemWithTag:`	Selects the menu item with the specified tag.

NSPopUpButtonCell

Complete reference information is available in the `NSPopUpButtonCell` reference.

Instance Methods

`selectItemWithTag:`	Selects the menu item with the specified tag.

NSPrintOperation

Complete reference information is available in the `NSPrintOperation` reference.

Instance Methods

`setShowsPrintPanel:`	Sets whether the receiver displays a print panel for this operation.
`setShowsProgressPanel:`	Sets whether the receiver displays a progress panel for this operation.
`showsPrintPanel`	Returns a Boolean value indicating whether a print panel is displayed during the operation,
`showsProgressPanel`	Returns a Boolean value indicating whether a progress panel is displayed during the operation.

NSResponder

Complete reference information is available in the `NSResponder` reference.

Instance Methods

`insertContainerBreak:`	Implemented by subclasses to insert a container break (typically a page break) at the insertion point or selection, deleting the selection if there is one.

insertLineBreak:	Implemented by subclasses to insert a line break (as distinguished from a paragraph break) at the insertion point or selection, deleting the selection if there is one.
presentError:	Presents an error alert to the user as an application-modal dialog.
presentError:modalForWindow:delegate: didPresentSelector:contextInfo:	Presents an error alert to the user as a document-modal sheet attached to document window.
tabletPoint:	Informs the receiver that a tablet-point event has occurred.
tabletProximity:	Informs the receiver that a tablet-proximity event has occurred.
willPresentError:	Implemented by subclasses to return a custom version of the supplied error object that is more suitable for presentation in alert sheets and dialogs.

NSScreen

Complete reference information is available in the NSScreen reference.

Instance Methods

userSpaceScaleFactor	Returns the scaling factor from user space to device space on the screen represented by the receiver.

NSSearchFieldCell

Complete reference information is available in the NSSearchFieldCell reference.

Instance Methods

sendsSearchStringImmediately	Returns a Boolean value indicating whether the receiver sends its action immediately upon being notified of changes to the search field text or after a brief pause.
setSendsSearchStringImmediately:	Sets whether the cell sends its action message to the target immediately upon notification of any changes to the search field text or after a brief pause.

NSSegmentedCell

Complete reference information is available in the NSSegmentedCell reference.

Instance Methods

selectSegmentWithTag:	Selects the segment with the specified tag.

NSSegmentedControl

Complete reference information is available in the NSSegmentedControl reference.

Instance Methods

selectSegmentWithTag:	Selects the segment with the specified tag.

NSStatusItem

Complete reference information is available in the NSStatusItem reference.

Instance Methods

doubleAction	Returns the selector that is sent to the receiver's target when the user double-clicks the receiver.
setDoubleAction:	Sets the selector that is sent to the receiver's target when the receiver is double-clicked.

NSString

Complete reference information is available in the NSString reference.

Instance Methods

boundingRectWithSize:options:attributes:	Calculates and returns the bounding rect for the receiver drawn using the given options and display characteristics, within the specified rectangle in the current graphics context.

drawWithRect:options:attributes:	Draws the receiver with the specified options and other display characteristics of the given attributes, within the specified rectangle in the current graphics context.

NSTableColumn

Complete reference information is available in the NSTableColumn reference.

Instance Methods

resizingMask	Returns the receiver's resizing mask.
setResizingMask:	Sets the resizing mask for the receiver to resizingMask.

NSTableView

Complete reference information is available in the NSTableView reference.

Instance Methods

canDragRowsWithIndexes:atPoint:	Returns whether the receiver allows dragging the rows at rowIndexes with a drag initiated at mousedDownPoint.
columnAutoresizingStyle	Returns the receiver's column autoresizing style.
dragImageForRowsWithIndexes:tableColumns: event:offset:	Computes and returns an image to use for dragging.
noteHeightOfRowsWithIndexesChanged:	Informs the receiver that the rows specified in indexSet have changed height.
setColumnAutoresizingStyle:	Sets the column autoresizing style of the receiver to a given style.
setDraggingSourceOperationMask:forLocal:	Sets the default operation mask returned by draggingSourceOperationMaskForLocal: to mask.

Delegate Methods

tableView:heightOfRow:	Returns the height of row in tableView.
tableView: namesOfPromisedFilesDroppedAtDestination: forDraggedRowsWithIndexes:	Returns an array of filenames that represent theindexSet rows for a drag to dropDestination.

`tableView:toolTipForCell:rect:tableColumn:row:mouseLocation:`	Returns a string that is displayed as a tooltip for aCell in aTableColumn of aTableView.
`tableView:writeRowsWithIndexes:toPasteboard:`	Returns a Boolean value that indicates whether a drag operation is allowed.

NSText

Complete reference information is available in the `NSText` reference.

Instance Methods

`baseWritingDirection`	Returns the initial writing direction used to determine the actual writing direction for text.
`setBaseWritingDirection:`	Sets the initial writing direction used to determine the actual writing direction for text.

NSTextBlock (New)

Complete reference information is available in the `NSTextBlock` reference.

Instance Methods

`backgroundColor`	Returns the background color of the text block.
`borderColorForEdge:`	Returns the border color of the specified text block edge.
`boundsRectForContentRect:inRect:textContainer:characterRange:`	Returns the rectangle the text in the block actually occupies, including padding, borders, and margins.
`contentWidth`	Returns the width of the text block.
`contentWidthValueType`	Returns the type of value stored for the text block width.
`drawBackgroundWithFrame:inView:characterRange:layoutManager:`	Called by the layout manager to draw any colors and other decorations before the text is drawn.
`init`	Initializes and returns an empty text block object.
`rectForLayoutAtPoint:inRect:textContainer:characterRange:`	Returns the rectangle within which glyphs should be laid out for the specified arguments.
`setBackgroundColor:`	Sets the background color of the text block.

setBorderColor:	Sets the color of all borders of the text block.
setBorderColor:forEdge:	Sets the border color of the specified edge of the text block.
setContentWidth:type:	Sets the width of the text block.
setValue:type:forDimension:	Sets a dimension of the text block.
setVerticalAlignment:	Sets the vertical alignment of the text block.
setWidth:type:forLayer:	Sets the width of all edges of a specified layer of the text block.
setWidth:type:forLayer:edge:	Sets the width of a specified edge of a specified layer of the text block.
valueForDimension:	Returns the value of the specified text block dimension.
valueTypeForDimension:	Returns the value type of the specified text block dimension.
verticalAlignment	Returns the vertical alignment of the text block.
widthForLayer:edge:	Returns the width of an edge of a specified layer of the text block.
widthValueTypeForLayer:edge:	Returns the value type of an edge of a specified layer of the text block.

NSTextList (New)

Complete reference information is available in the NSTextList reference.

Instance Methods

initWithMarkerFormat:options:	Returns an initialized text list.
listOptions	Returns the list options mask value of the receiver.
markerForItemNumber:	Returns the computed value for a specific ordinal position in the list.
markerFormat	Returns the marker format string used by the receiver.

NSTextTable (New)

Complete reference information is available in the NSTextTable reference.

Instance Methods

`boundsRectForBlock:contentRect:inRect:textContainer:characterRange:`	Returns the rectangle the text table block actually occupies, including padding, borders, and margins.
`collapsesBorders`	Returns whether the text table borders are collapsible.
`drawBackgroundForBlock:withFrame:inView:characterRange:layoutManager:`	Draws any colors and other decorations for a text table block.
`hidesEmptyCells`	Returns whether the text table hides empty cells.
`layoutAlgorithm`	Returns the text table layout algorithm.
`numberOfColumns`	Returns the number of columns in the text table.
`rectForBlock:layoutAtPoint:inRect:textContainer:characterRange:`	Returns the rectangle within which glyphs should be laid out for a text table block.
`setCollapsesBorders:`	Sets whether the text table borders are collapsible.
`setHidesEmptyCells:`	Sets whether the text table hides empty cells.
`setLayoutAlgorithm:`	Sets the text table layout algorithm.
`setNumberOfColumns:`	Sets the number of columns in the text table.

NSTextTableBlock (New)

Complete reference information is available in the `NSTextTableBlock` reference.

Instance Methods

`columnSpan`	Returns the number of table columns spanned by this text table block.
`initWithTable:startingRow:rowSpan:startingColumn:columnSpan:`	Returns an initialized text table block.
`rowSpan`	Returns the number of table rows spanned by this text table block.
`startingColumn`	Returns the table column at which this text table block starts.
`startingRow`	Returns the table row at which this text table block starts.
`table`	Returns the table containing this text table block.

NSTextView

Complete reference information is available in the NSTextView reference.

Instance Methods

breakUndoCoalescing	Informs the receiver that it should begin coalescing successive typing operations in a new undo grouping.
orderFrontLinkPanel:	Brings forward a panel allowing the user to manipulate links in the text view.
orderFrontListPanel:	Brings forward a panel allowing the user to manipulate text lists in the text view.
orderFrontSpacingPanel:	Brings forward a panel allowing the user to manipulate text line heights, interline spacing, and paragraph spacing, in the text view.
orderFrontTablePanel:	Brings forward a panel allowing the user to manipulate text tables in the text view.
rangesForUserCharacterAttributeChange	Returns an array containing the ranges of characters affected by an action method that changes character (not paragraph) attributes.
rangesForUserParagraphAttributeChange	Returns an array containing the ranges of characters affected by a method that changes paragraph (not character) attributes.
rangesForUserTextChange	Returns an array containing the ranges of characters affected by a method that changes characters (as opposed to attributes).
selectedRanges	Returns an array containing the ranges of characters selected in the receiver’s layout manager.
setBaseWritingDirection:range:	Sets the base writing direction of a range of text.
setSelectedRanges:	Sets the selection to the characters in an array of ranges.
setSelectedRanges:affinity:stillSelecting:	Sets the selection to the characters in an array of ranges in response to user action.
shouldChangeTextInRanges:replacementStrings:	Initiates a series of delegate messages (and general notifications) to determine whether modifications can be made to the characters and attributes of the receiver’s text.

Delegate Methods

textView:shouldChangeTextInRanges: replacementStrings:	Sent when a text view needs to determine if text in an array of specified ranges should be changed.
textView:shouldChangeTypingAttributes: toAttributes:	Sent when the typing attributes are changed.
textView:willChangeSelectionFromCharacterRanges: toCharacterRanges:	Returns the actual character ranges to select.

NSTokenField (New)

Complete reference information is available in the NSTokenField reference.

Class Methods

defaultCompletionDelay	Returns the default completion delay.
defaultTokenizingCharacterSet	Returns the default tokenizing character set.

Instance Methods

completionDelay	Returns the receiver's completion delay.
setCompletionDelay:	Sets the receiver's completion delay.
setTokenizingCharacterSet:	Sets the recevier's tokenizing character set to characterSet.
setTokenStyle:	Returns the token style of the receiver.
tokenizingCharacterSet	Returns the receiver's tokenizing character set.
tokenStyle	Returns the receiver's token style.

Delegate Methods

tokenField:completionsForSubstring: indexOfToken:indexOfSelectedItem:	Allows the delegate to provide an array of appropriate completions for the contents of the receiver.
tokenField:displayStringForRepresentedObject:	Allows the delegate to provide a string to be displayed as a proxy for the given represented object.

tokenField:editingStringForRepresentedObject:	Allows the delegate to provide a string to be edited as a proxy for a represented object.
tokenField:hasMenuForRepresentedObject:	Allows the delegate to specify whether the given represented object provides a menu.
tokenField:menuForRepresentedObject:	Allows the delegate to provide a menu for the specified represented object.
tokenField:readFromPasteboard:	Allows the delegate to return an array of objects representing the data read from the specified pasteboard.
tokenField:representedObjectForEditingString:	Allows the delegate to provide a represented object for the given editing string.
tokenField:shouldAddObjects:atIndex:	Allows the delegate to validate the tokens to be added to the receiver at a particular location.
tokenField:styleForRepresentedObject:	Allows the delegate to return the token style for editing the specified represented object.
tokenField:writeRepresentedObjects:toPasteboard:	Sent so the delegate can write represented objects to the pasteboard corresponding to a given array of display strings.

NSTokenFieldCell (New)

Complete reference information is available in the NSTokenFieldCell reference.

Class Methods

defaultCompletionDelay	Returns the default completion delay.
defaultTokenizingCharacterSet	Returns the default tokenizing character set.

Instance Methods

completionDelay	Returns the receiver’s completion delay.
delegate	Returns the receiver’s delegate.
setCompletionDelay:	Sets the receiver’s completion delay to a given delay.

`setDelegate:`	Sets the receiver's delegate.
`setTokenizingCharacterSet:`	Sets the receiver's tokenizing character set to a given character set.
`setTokenStyle:`	Sets the token style of the receiver.
`tokenizingCharacterSet`	Returns the receiver's tokenizing character set.
`tokenStyle`	Returns the receiver's token style.

Delegate Methods

`tokenFieldCell:completionsForSubstring:indexOfToken:indexOfSelectedItem:`	Allows the delegate to provide an array of appropriate completions for the contents of the receiver.
`tokenFieldCell:displayStringForRepresentedObject:`	Allows the delegate to provide a string to be displayed as a proxy for the represented object.
`tokenFieldCell:editingStringForRepresentedObject:`	Allows the delegate to provide a string to be edited as a proxy for the represented object.
`tokenFieldCell:hasMenuForRepresentedObject:`	Allows the delegate to specify whether the represented object provides a menu.
`tokenFieldCell:menuForRepresentedObject:`	Allows the delegate to provide a menu for the specified represented object.
`tokenFieldCell:readFromPasteboard:`	Allows the delegate to return an array of objects representing the data read from pboard.
`tokenFieldCell:representedObjectForEditingString:`	Allows the delegate to provide a represented object for the string being edited.
`tokenFieldCell:shouldAddObjects:atIndex:`	Allows the delegate to validate the tokens to be added to the receiver at a given index.
`tokenFieldCell:styleForRepresentedObject:`	Allows the delegate to return the token style for editing the specified represented object.
`tokenFieldCell:writeRepresentedObjects:toPasteboard:`	Allows the delegate the opportunity to write custom pasteboard types to the pasteboard for the represented objects in objects.

NSToolbar

Complete reference information is available in the NSToolbar reference.

Instance Methods

setShowsBaselineSeparator:	Sets whether the toolbar shows the separator between the toolbar and the main window contents.
showsBaselineSeparator	Returns a Boolean value that indicates whether the toolbar shows the separator between the toolbar and the main window contents.

NSToolbarItem

Complete reference information is available in the NSToolbarItem reference.

Instance Methods

autovalidates	Returns a Boolean value that indicates whether the receiver is automatically validated by the toolbar.
setAutovalidates:	Sets the receiver's auto validation flag.
setVisibilityPriority:	Sets the receiver's visibility priority.
visibilityPriority	Returns the receiver's visibility priority.

NSTreeController (New)

Complete reference information is available in the NSTreeController reference.

Instance Methods

add:	Adds an object to the receiver after the current selection.
addChild:	Adds a child object to the currently selected item.
addSelectionIndexPaths:	Adds the objects at the specified indexPaths in the receiver's content to the current selection.
alwaysUsesMultipleValuesMarker	Returns whether the receiver always returns the multiple values marker when multiple objects are selected, even if the selected items have the same value.

`arrangedObjects`	Returns a proxy root tree node containing the receiver’s sorted content objects.
`avoidsEmptySelection`	Returns whether the receiver requires that the content array attempt to maintain a selection at all times.
`canAddChild`	Returns YES if a child object can be added to the receiver’s content.
`canInsert`	Returns YES if an object can be inserted into the receiver’s content.
`canInsertChild`	Returns YES if a child object can be inserted into the receiver’s content.
`childrenKeyPath`	Returns the key path used to find the children in the receiver’s objects.
`content`	Returns the receiver’s content object.
`countKeyPath`	Returns the key path used to find the number of children for a node.
`insert:`	Creates a new object of the class specified by objectClass and inserts it into the receiver’s content.
`insertChild:`	Creates a new object of the class specified by objectClass and inserts it into the receiver’s content as a child of the current selection.
`insertObject:atArrangedObjectIndexPath:`	Inserts object into the receiver’s arranged objects array at the location specified by indexPath, and adds it to the receiver’s content.
`insertObjects:atArrangedObjectIndexPaths:`	Inserts objects into the receiver’s arranged objects array at the locations specified in indexPaths, and adds them to the receiver’s content.
`leafKeyPath`	Returns the key path used by the receiver to determine if a node is a leaf key.
`preservesSelection`	Returns whether the receiver will attempt to preserve the current selection when the content changes.
`rearrangeObjects`	Use this method to trigger reordering of the receiver’s content.
`remove:`	Removes the receiver’s selected objects from the content.

removeObjectAtArrangedObjectIndexPath:	Removes the object at the specified indexPath in the receiver's arranged objects from the receiver's content.
removeObjectsAtArrangedObjectIndexPaths:	Removes the objects at the specified indexPaths in the receiver's arranged objects from the receiver's content.
removeSelectionIndexPaths:	Removes the objects at the specified indexPaths from the receiver's current selection, returning YES if the selection was changed.
selectedObjects	Returns an array containing the receiver's selected objects.
selectionIndexPath	Returns the index path of the first object in the receiver's selection, or nil if there is no selection.
selectionIndexPaths	Returns an array containing the index paths of the receiver's currently selected objects in the content.
selectsInsertedObjects	Returns whether the receiver selects inserted objects automatically.
setAlwaysUsesMultipleValuesMarker:	Sets whether the receiver always returns the multiple values marker when multiple objects are selected, even if they have the same value.
setAvoidsEmptySelection:	Sets whether the receiver will attempt to avoid an empty selection.
setChildrenKeyPath:	Sets the key path used by the receiver to access child objects to key.
setContent:	Sets the receiver's content to content.
setCountKeyPath:	Sets the key path used by the receiver to determine the number of objects at a node to key.
setLeafKeyPath:	Sets the key path used by the receiver to determine if an object is a leaf node to key.
setPreservesSelection:	Sets whether the receiver will attempt to preserve selection when the content changes.
setSelectionIndexPath:	Sets the receiver's current selection to indexPath, returning YES if the selection was changed.
setSelectionIndexPaths:	Sets the receiver's current selection to indexPaths, returning YES if the selection was changed.
setSelectsInsertedObjects:	Sets whether the receiver will automatically select objects as they are inserted.

`setSortDescriptors:`	Sets the sort descriptors used to arrange the receiver's contents.
`sortDescriptors`	Returns an array containing the sort descriptors used to arrange the receiver's content.

NSTypesetter

Complete reference information is available in the `NSTypesetter` reference.

Instance Methods

`actionForControlCharacterAtIndex:`	Returns the action associated with a control character.
`attributedString`	Returns the text backing store, usually an instance of NSTextStorage.
`attributesForExtraLineFragment`	Returns the attributes used to lay out the extra line fragment.
`beginLineWithGlyphAtIndex:`	Sets up layout parameters at the beginning of a line during typesetting.
`beginParagraph`	Sets up layout parameters at the beginning of a paragraph.
`bidiProcessingEnabled`	Returns whether bidirectional text processing is enabled.

`boundingBoxForControlGlyphAtIndex:`
`forTextContainer:proposedLineFragment:`
`glyphPosition:characterIndex:`
	Returns the bounding rectangle for the specified control glyph with the specified parameters.

`characterRangeForGlyphRange:actualGlyphRange:`	Returns the range for the characters in the receiver's text store that are mapped to the specified glyphs.
`currentParagraphStyle`	Returns the paragraph style object for the text being typeset.
`currentTextContainer`	Returns the text container for the text being typeset.
`deleteGlyphsInRange:`	Deletes the specified glyphs from the glyph cache maintained by the layout manager.
`endLineWithGlyphRange:`	Sets up layout parameters at the end of a line during typesetting.

`endParagraph`	Sets up layout parameters at the end of a paragraph.
`getGlyphsInRange:glyphs:characterIndexes:glyphInscriptions:elasticBits:bidiLevels:`	Extracts the information needed to lay out the provided glyphs from the provided range.
`getLineFragmentRect:usedRect:forParagraphSeparatorGlyphRange:atProposedOrigin:`	Calculates the line fragment rectangle and line fragment used rectangle for blank lines.
`getLineFragmentRect:usedRect:remainingRect:forStartingGlyphAtIndex:proposedRect:lineSpacing:paragraphSpacingBefore:paragraphSpacingAfter:`	Calculates line fragment rectangle, line fragment used rectangle, and remaining rectangle for a line fragment.
`glyphRangeForCharacterRange:actualCharacterRange:`	Returns the range for the glyphs mapped to the characters of the text store in the specified range.
`hyphenationFactor`	Returns the current hyphenation factor.
`hyphenationFactorForGlyphAtIndex:`	Returns the hyphenation factor in effect at a specified location.
`hyphenCharacterForGlyphAtIndex:`	Returns the hyphen character to be inserted after the specified glyph.
`insertGlyph:atGlyphIndex:characterIndex:`	Enables the typesetter to insert a new glyph into the stream.
`layoutManager`	Returns the layout manager for the text being typeset.
`layoutParagraphAtPoint:`	Lays out glyphs in the current glyph range until the next paragraph separator is reached.
`lineFragmentPadding`	Returns the current line fragment padding, in points.
`lineSpacingAfterGlyphAtIndex:withProposedLineFragmentRect:`	Returns the line spacing in effect following the specified glyph.
`paragraphCharacterRange`	Returns the character range currently being processed.
`paragraphGlyphRange`	Returns the glyph range currently being processed.
`paragraphSeparatorCharacterRange`	Returns the current paragraph separator character range.

paragraphSeparatorGlyphRange	Returns the current paragraph separator range.
paragraphSpacingAfterGlyphAtIndex: withProposedLineFragmentRect:	Returns the paragraph spacing that is in effect after the specified glyph.
paragraphSpacingBeforeGlyphAtIndex: withProposedLineFragmentRect:	Returns the number of points of space—added before a paragraph—that is in effect before the specified glyph.
setAttachmentSize:forGlyphRange:	Sets the size the specified glyphs (assumed to be attachments) will be asked to draw themselves at.
setAttributedString:	Sets the text backing store on which this typesetter operates.
setBidiLevels:forGlyphRange:	Sets the direction of the specified glyphs for bidirectional text.
setBidiProcessingEnabled:	Controls whether the typesetter performs bidirectional text processing.
setDrawsOutsideLineFragment:forGlyphRange:	Sets whether the specified glyphs exceed the bounds of the line fragment in which they are laid out.
setHardInvalidation:forGlyphRange:	Sets whether to force the layout manager to invalidate the specified portion of the glyph cache when invalidating layout.
setHyphenationFactor:	Sets the threshold controlling when hyphenation is attempted.
setLineFragmentPadding:	Sets the amount (in points) by which text is inset within line fragment rectangles.
setLineFragmentRect:forGlyphRange: usedRect:baselineOffset:	Sets the line fragment rectangle where the specified glyphs are laid out.
setLocation:withAdvancements:forStartOfGlyphRange:	Sets the location where the specified glyphs are laid out.
setNotShownAttribute:forGlyphRange:	Sets whether the specified glyphs are not shown.
setParagraphGlyphRange:separatorGlyphRange:	Sets the current glyph range being processed.
setTypesetterBehavior:	Sets the default typesetter behavior, which affects glyph spacing and line height.

setUsesFontLeading:	Sets whether the typesetter uses the leading (or line gap) value specified in the font metric information.
shouldBreakLineByHyphenatingBeforeCharacterAtIndex:	Returns whether the line being laid out should be broken by hyphenating at the specified character.
shouldBreakLineByWordBeforeCharacterAtIndex:	Returns whether the line being laid out should be broken by a word break at the specified character.
substituteFontForFont:	Returns a screen font suitable for use in place of a given font.
substituteGlyphsInRange:withGlyphs:	Replaces the specified glyphs with specified replacement glyphs.
textContainers	Returns an array containing the text containers belonging to the current layout manager.
textTabForGlyphLocation:writingDirection: maxLocation:	Returns the text tab next closest to a given glyph location within the given parameters.
typesetterBehavior	Returns the current typesetter behavior.
usesFontLeading	Returns whether the typesetter uses the leading (or line gap) value specified in the font metric information of the current font.
willSetLineFragmentRect:forGlyphRange: usedRect:baselineOffset:	Called by the typesetter just prior to storing the actual line fragment rectangle location in the layout manager.

NSUserDefaultsController

Complete reference information is available in the NSUserDefaultsController reference.

Instance Methods

hasUnappliedChanges	Returns whether the receiver has user default values that have not been saved to NSUserDefaults.

NSView

Complete reference information is available in the `NSView` reference.

Instance Methods

`bitmapImageRepForCachingDisplayInRect:`	Returns a bitmap-representation object suitable for caching the specified portion of the receiver.
`cacheDisplayInRect:toBitmapImageRep:`	Draws the specified area of the receiver, and its descendants, into a provided bitmap-representation object.
`displayRectIgnoringOpacity:inContext:`	Causes the receiver and its descendants to be redrawn to the specified graphics context.
`getRectsExposedDuringLiveResize:count:`	Returns a list of rectangles indicating the newly exposed areas of the receiver.
`lockFocusIfCanDrawInContext:`	Locks the focus to the receiver atomically if drawing can occur in the specified graphics context.
`pageFooter`	Returns a default footer string that includes the current page number and page count.
`pageHeader`	Returns a default header string that includes the print job title and date.
`preservesContentDuringLiveResize`	Returns YES if the view supports the optimization of live-resize operations by preserving content that has not moved; otherwise, returns NO.
`rectPreservedDuringLiveResize`	Returns the rectangle identifying the portion of your view that did not change during a live resize operation.
`registeredDraggedTypes`	Returns the array of pasteboard drag types that the view can accept.

NSViewAnimation (New)

Complete reference information is available in the `NSViewAnimation` reference.

Instance Methods

`initWithViewAnimations:`	Returns an NSViewAnimation object initialized with the supplied information.
`setViewAnimations:`	Sets the dictionaries defining the objects to animate.
`viewAnimations`	Returns the array of dictionaries defining the objects to animate.

NSWindow

Complete reference information is available in the NSWindow reference.

Instance Methods

autorecalculatesKeyViewLoop	Indicates whether the window automatically recalculates the key view loop when views are added.
disableScreenUpdatesUntilFlush	Disables the window's screen updates until the window is flushed.
displaysWhenScreenProfileChanges	Indicates whether the window context should be updated when the screen profile changes or when the window moves to a different screen.
graphicsContext	Provides the graphics context associated with the window for the current thread.
preservesContentDuringLiveResize	Returns a Boolean value that indicates whether the window tries to optimize live resize operations by preserving the content of views that have not changed.
recalculateKeyViewLoop	Marks the key view loop as dirty and in need of recalculation.
setAutorecalculatesKeyViewLoop:	Specifies whether to recalculate the key view loop automatically when views are added or removed.
setDisplaysWhenScreenProfileChanges:	Specifies whether the window context should be updated when the screen profile changes.
setPreservesContentDuringLiveResize:	Specifies whether the window tries to optimize live resize operations by preserving the content of views that have not changed.
setShowsToolbarButton:	Specifies whether the window shows the toolbar control button.
showsToolbarButton	Indicates whether the toolbar control button is currently displayed.
userSpaceScaleFactor	Returns the scale factor applied to the window.

NSWorkspace

Complete reference information is available in the NSWorkspace reference.

Instance Methods

`setIcon:forFile:options:`	Sets the icon for the file or directory at the specified path.

C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

AppKitErrors.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSTextReadInapplicableDocumentTypeError`	Indicates a problem reading data with the specified format.
`NSTextReadWriteErrorMaximum`	The end of a range of error codes reserved for future use.
`NSTextReadWriteErrorMinimum`	The beginning of a range of error codes reserved for future use.
`NSTextWriteInapplicableDocumentTypeError`	Indicates a problem writing data of the specified format.

NSAccessibility.h

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSAccessibilityActionDescription`	Returns a standard description for an action.
`NSAccessibilityRoleDescription`	Returns a standard description for a role and subrole.
`NSAccessibilityRoleDescriptionForUIElement`	Returns a standard role description for a user interface element.

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAccessibilityAllowedValuesAttribute	Array of allowed values (NSArray)
NSAccessibilityAscendingSortDirectionValue	Column sorted in ascending values
NSAccessibilityAttachmentTextAttribute	Text attachment (id)
NSAccessibilityAttributedString-ForRangeParameterizedAttribute	Does not use attributes from Appkit/AttributedString.h (NSAttributedString)
NSAccessibilityBackgroundColorTextAttribute	Text background color (CGColorRef)
NSAccessibilityCenterTabStopMarkerTypeValue	Center tab stop
NSAccessibilityCentimetersUnitValue	Ruler units are centimeters.
NSAccessibilityCreatedNotification	Element has been created.
NSAccessibilityDecimalTabStopMarkerTypeValue	Decimal tab stop
NSAccessibilityDeleteAction	Deletes the value of object.
NSAccessibilityDescendingSortDirectionValue	Column sorted in descending values
NSAccessibilityDescriptionAttribute	Instance description (NSString)
NSAccessibilityDisclosureLevelAttribute	Indentation level of this row (NSNumber)
NSAccessibilityFirstLineIndentMarkerTypeValue	First line indent marker
NSAccessibilityFontFamilyKey	Optional key for font family
NSAccessibilityFontNameKey	Required key for font name
NSAccessibilityFontSizeKey	Required key for font size
NSAccessibilityFontTextAttribute	Font keys (NSDictionary)
NSAccessibilityForegroundColorTextAttribute	Text foreground color (CGColorRef)
NSAccessibilityHeadIndentMarkerTypeValue	Head indent marker
NSAccessibilityHelpTagCreatedNotification	A help tag has appeared.
NSAccessibilityHelpTagRole	Help tag
NSAccessibilityInchesUnitValue	Ruler units are inches.
NSAccessibilityIndexAttribute	Index of element (NSValue)

`NSAccessibilityInsertionPointLineNumberAttribute`	Line number containing caret (NSNumber)
`NSAccessibilityLabelUIElementsAttribute`	Array of label elements (NSArray)
`NSAccessibilityLabelValueAttribute`	Value of label element (NSNumber)
`NSAccessibilityLeftTabStopMarkerTypeValue`	Left tab stop
`NSAccessibilityLinkedUIElementsAttribute`	Elements corresponding to each other (NSArray)
`NSAccessibilityLinkRole`	Link
`NSAccessibilityLinkTextAttribute`	Text link (id)
`NSAccessibilityMarkerGroupUIElementAttribute`	Marker group user interface element (id)
`NSAccessibilityMarkerTypeAttribute`	Type of marker (NSString). See "Ruler marker type values".
`NSAccessibilityMarkerTypeDescriptionAttribute`	Description of marker type (NSString)
`NSAccessibilityMarkerUIElementsAttribute`	Array of marker user interface elements (NSArray)
`NSAccessibilityMarkerValuesAttribute`	Marker values
`NSAccessibilityMatteContentUIElementAttribute`	Element clipped by the matte (id)
`NSAccessibilityMatteHoleAttribute`	Bounds of matte hole in screen coordinates (NSValue containing an NSRect)
`NSAccessibilityMatteRole`	Matte
`NSAccessibilityMisspelledTextAttribute`	Misspelled text (NSNumber as a Boolean value)
`NSAccessibilityMovedNotification`	Element has been moved.
`NSAccessibilityPicasUnitValue`	Ruler units are picas.
`NSAccessibilityPointsUnitValue`	Ruler units are points.
`NSAccessibilityResizedNotification`	Element has been resized.
`NSAccessibilityRightTabStopMarkerTypeValue`	Right tab stop
`NSAccessibilityRowCountChangedNotification`	A row has been added or deleted.
`NSAccessibilityRulerMarkerRole`	Ruler marker
`NSAccessibilityRulerRole`	Ruler

NSAccessibilitySelectedChildrenChangedNotification	Selected child elements have changed.
NSAccessibilitySelectedColumnsChangedNotification	Selected columns have changed.
NSAccessibilitySelectedRowsChangedNotification	Selected rows have changed.
NSAccessibilitySelectedTextChangedNotification	Selected text has changed.
NSAccessibilityServesAsTitleForUIElementsAttribute	Elements to which title applies (NSArray)
NSAccessibilityShadowTextAttribute	Text shadow (NSNumber as a Boolean value)
NSAccessibilitySharedCharacterRangeAttribute	The (rangeValue) part of shared text in this view (NSValue)
NSAccessibilitySharedTextUIElementsAttribute	Text views sharing text (NSArray)
NSAccessibilityShowMenuAction	Simulates showing a menu by clicking on it.
NSAccessibilityShownMenuAttribute	Menu currently being displayed (id)
NSAccessibilitySortButtonRole	Sort button
NSAccessibilityStrikethroughColorTextAttribute	Text stikethrough color (CGColorRef)
NSAccessibilityStrikethroughTextAttribute	Text strikethrough (NSNumber as a Boolean value)
NSAccessibilitySuperscriptTextAttribute	Text superscript style (NSNumber). Values > 0 are superscript; values < 0 are subscript.
NSAccessibilityTailIndentMarkerTypeValue	Tail indent marker
NSAccessibilityTextAttachmentSubrole	Text attachment
NSAccessibilityTextLinkSubrole	Text link
NSAccessibilityTitleChangedNotification	Title has changed.
NSAccessibilityTopLevelUIElementAttribute	User interface element for the top-level element (id)
NSAccessibilityUnderlineColorTextAttribute	Text underline color (CGColorRef)
NSAccessibilityUnderlineTextAttribute	Text underline style (NSNumber)
NSAccessibilityUnitDescriptionAttribute	Description of ruler units (NSString)
NSAccessibilityUnitsAttribute	Ruler units (NSString). See "Ruler unit attributes".
NSAccessibilityUnknownMarkerTypeValue	Unknown marker type

NSAccessibilityUnknownSortDirectionValue	Sort direction unknown
NSAccessibilityUnknownUnitValue	Ruler units are unknown.
NSAccessibilityURLAttribute	URL associated with element (NSURL)
NSAccessibilityVisibleNameKey	Optional key for font visibility

NSAnimation.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAnimationBlocking	Requests the animation to run in the main thread in a custom run-loop mode that blocks user input.
NSAnimationBlockingMode	These constants indicate the blocking mode of an NSAnimation object when it is running.
NSAnimationCurve	These constants describe the curve of an animation—that is, the relative speed of an animation from start to finish.
NSAnimationEaseIn	Describes an animation that slows down as it reaches the end.
NSAnimationEaseInOut	Describes an S-curve in which the animation slowly speeds up and then slows down near the end of the animation. This constant is the default.
NSAnimationEaseOut	Describes an animation that slowly speeds up from the start.
NSAnimationLinear	Describes an animation in which there is no change in frame rate.
NSAnimationNonblocking	Requests the animation to run in a standard or specified run-loop mode that allows user input.
NSAnimationNonblockingThreaded	Requests the animation to run in a separate thread that is spawned by the NSAnimation object.
NSAnimationProgress	
NSAnimationProgressMark	Contains a float as an NSNumber instance that indicates the current animation progress.
NSAnimationProgressMarkNotification	Posted when the current progress of a running animation reaches one of its progress marks.

NSViewAnimationEffectKey	An effect to apply to the animation.
NSViewAnimationEndFrameKey	The size and location of the window or view at the end of the animation.
NSViewAnimationFadeInEffect	Specifies a fade-in type of effect.
NSViewAnimationFadeOutEffect	Specifies a fade-out type of effect.
NSViewAnimationStartFrameKey	The size and location of the window or view at the start of the animation.
NSViewAnimationTargetKey	The target of the animation.

NSApplication.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAppKitVersionNumber10_3	The Application Kit framework included in Mac OS X v10.3.
NSAppKitVersionNumber10_3_2	The Application Kit framework included in Mac OS X v10.3.2.
NSAppKitVersionNumber10_3_3	The Application Kit framework included in Mac OS X v10.3.3.
NSAppKitVersionNumber10_3_5	The Application Kit framework included in Mac OS X v10.3.5.
NSApplicationPrintReply	These constants are returned by application:printFiles: withSettings:showPrintPanels: .
NSPrintingCancelled	Printing was cancelled.
NSPrintingFailure	Printing failed.
NSPrintingReplyLater	The result of printing cannot be returned immediately, for example, if printing will cause the presentation of a sheet.
NSPrintingSuccess	Printing was successful.

NSAttributedString.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAuthorDocumentAttribute	NSString containing author name.
NSBackgroundColorDocumentAttribute	@"BackgroundColor"
NSBaseURLDocumentOption	@"BaseURL"
NSBottomMarginDocumentAttribute	@"BottomMargin"
NSCharacterEncodingDocumentAttribute	@"CharacterEncoding"
NSCharacterEncodingDocumentOption	@"CharacterEncoding"
NSCocoaVersionDocumentAttribute	@"CocoaRTFVersion"
NSCommentDocumentAttribute	NSString containing document comments.
NSCompanyDocumentAttribute	NSString containing company or organization name.
NSConvertedDocumentAttribute	@"Converted"
NSCopyrightDocumentAttribute	NSString containing document copyright info.
NSCreationTimeDocumentAttribute	NSDate containing the creation date of the document; note that this is not the file system creation date of the file, but of the document.
NSDefaultAttributesDocumentOption	@"DefaultAttributes"
NSDefaultTabIntervalDocumentAttribute	@"DefaultTabInterval"
NSDocumentTypeDocumentAttribute	@"DocumentType"
NSDocumentTypeDocumentOption	@"DocumentType"
NSEditorDocumentAttribute	NSString containing name of person who last edited the document.
NSExcludedElementsDocumentAttribute	An NSArray object containing NSString objects, representing HTML elements not to be used in generated HTML.
NSHyphenationFactorDocumentAttribute	@"HyphenationFactor"
NSKeywordsDocumentAttribute	NSArray of NSString, containing keywords.
NSLeftMarginDocumentAttribute	@"LeftMargin"
NSModificationTimeDocumentAttribute	NSDate containing the modification date of the document contents.
NSPaperSizeDocumentAttribute	@"PaperSize"
NSPrefixSpacesDocumentAttribute	An NSNumber containing an integer (default 0) representing the number of spaces per level by which to indent certain nested HTML elements.

NSReadOnlyDocumentAttribute	@"ReadOnly"
NSRightMarginDocumentAttribute	@"RightMargin"
NSSubjectDocumentAttribute	NSString containing subject of document.
NSTextEncodingNameDocumentAttribute	An NSString object containing the name, IANA or otherwise, of a text encoding to be used; mutually exclusive with NSCharacterEncodingDocumentAttribute.
NSTextEncodingNameDocumentOption	@"TextEncodingName"
NSTextSizeMultiplierDocumentOption	Specifies a scale factor for font sizes.
NSTimeoutDocumentOption	@"Timeout"
NSTitleDocumentAttribute	NSString containing document title.
NSTopMarginDocumentAttribute	@"TopMargin"
NSViewModeDocumentAttribute	@"ViewMode"
NSViewSizeDocumentAttribute	@"ViewSize"
NSViewZoomDocumentAttribute	@"ViewZoom"
NSWebArchiveTextDocumentType	Web Kit WebArchive document.
NSWebPreferencesDocumentOption	@"WebPreferences"
NSWebResourceLoadDelegateDocumentOption	@"WebResourceLoadDelegate"

NSBitmapImageRep.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAlphaFirstBitmapFormat	If 0, alpha values are the last component.
NSAlphaNonpremultipliedBitmapFormat	If 0, alpha values are premultiplied.
NSBitmapFormat	These constants represent the various bitmap component formats supported by NSBitmapImageRep. These values are combined using the C bitwise OR operator and passed to initWithBitmapDataPlanes: pixelsWide:pixelsHigh: bitsPerSample:samplesPerPixel: hasAlpha:isPlanar:colorSpaceName: bitmapFormat:bytesPerRow: bitsPerPixel: as the bitmap format and are returned by bitmapFormat.

NSFloatingPointSamplesBitmapFormat	If 0, samples are integer values.
NSImageEXIFData	Identifies an NSDictionary object containing the EXIF data for the image.
NSImageGamma	Identifies an NSNumber object containing the gamma value for the image.
NSImageProgressive	Identifies an NSNumber object containing a boolean that indicates whether the image uses progressive encoding.
NSJPEG2000FileType	JPEG 2000 file format.

NSButtonCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSRecessedBezelStyle	A bezel style that matches the recessed buttons in Mail, Finder and Safari.
NSRoundedDisclosureBezelStyle	A bezel style that matches the disclosure style used in the standard Save panel.
NSRoundRectBezelStyle	A bezel style that matches the search buttons in Finder and Mail.
NSSmallSquareBezelStyle	A simple square bezel style. Buttons using this style can be scaled to any size.
NSTexturedRoundedBezelStyle	A textured (metal) bezel style similar in appearance to the Finder's action (gear) button.

NSColorSpace.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSCMYKColorSpaceModel	The CYMK (cyan, yellow, magenta, black) color-space model.
NSColorSpaceModel	The type of the color-space mode constants listed in "Color Space Models."
NSDeviceNColorSpaceModel	DeviceN is a color-space model from Adobe Systems, Inc.

NSGrayColorSpaceModel	The grayscale color-space model.
NSLABColorSpaceModel	The L*a*b* device-independent color-space model, which represents colors relative to a reference white point.
NSRGBColorSpaceModel	The RGB (red green blue) color-space model.
NSUnknownColorSpaceModel	This model is not known to NSColorSpace.

NSDatePickerCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSClockAndCalendarDatePickerStyle	Provide a visual clock and calendar style interface.
NSDatePickerElementFlags	Specifies a type for constants that allow you to specify the date and time elements that the NSDatePickerCell can edit.
NSDatePickerMode	Specifies a type for constants that define whether the control provides a single date, or a range of dates.
NSDatePickerStyle	Specifies a type for constants that define the visual appearance of the NSDatePickerCell.
NSEraDatePickerElementFlag	Display and allow editing of the era of the date, if applicable.
NSHourMinuteDatePickerElementFlag	Display and allow editing of the hour and minute elements of the date.
NSHourMinuteSecondDatePickerElementFlag	Display and allow editing of the hour, minute and second elements of the date.
NSRangeDateMode	Allow selection of a range of dates.
NSSingleDateMode	Allow selection of a single date.
NSTextFieldAndStepperDatePickerStyle	Provide a text field and stepper style interface.
NSTimeZoneDatePickerElementFlag	Display and allow editing of the time zone.
NSYearMonthDatePickerElementFlag	Display and allow editing of the year and month elements of the date.
NSYearMonthDayDatePickerElementFlag	Display and allow editing of the year, month and day elements of the date.

NSDocument.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSAutosaveOperation`	Specifies an autosave operation, writing a document’s contents to a file or file package separate from the document's current one, without changing the document's current one.
`NSChangeAutosaved`	The document's contents have been autosaved. For example, saveToURL:ofType:forSaveOperation: error: passes this value for a successful NSAutosaveOperation.
`NSChangeReadOtherContents`	The document has been initialized with the contents of a file or file package other than the one whose location would be returned by fileURL, and therefore can't possibly be synchronized with its persistent representation. For example, initForURL:withContentsOfURL: ofType:error: passes this value when the two passed-in URLs are not equal to indicate that an autosaved document is being reopened.

NSEvent.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSCursorPointingDevice`	Represents a cursor (or puck-like) pointing device.
`NSDeviceIndependentModifierFlagsMask`	Used to retrieve only the device-independent modifier flags, allowing applications to mask off the device-dependent modifier flags, including event coalescing information.
`NSEraserPointingDevice`	Represents the eraser end of a stylus-like pointing device.
`NSMouseEventSubtype`	Indicates a purely mouse event.
`NSPenLowerSideMask`	The button on the lower side of the device is activated.
`NSPenPointingDevice`	Represents the tip end of a stylus-like pointing device.
`NSPenTipMask`	The pen tip is activated.
`NSPenUpperSideMask`	The button on the upper side of the device is activated.

NSPointingDeviceType	The following constants represent pointing-device types for NSTabletProximity events or mouse events with subtype NSTabletProximityEventSubtype. The pointingDeviceType method returns one of these constants.
NSTabletPoint	An event representing the current state of a tablet pointing device, including its location, pressure, and tilt.
NSTabletPointEventSubtype	Indicates a tablet-pointer event; see description of NSTabletPoint.
NSTabletPointMask	Corresponds to NSTabletPoint.
NSTabletProximity	An event representing the proximity of a pointing device to its tablet.
NSTabletProximityEventSubtype	Indicates a tablet-proximity event; see description of NSTabletProximity.
NSTabletProximityMask	Corresponds to NSTabletProximity.
NSUnknownPointingDevice	Represents an unknown type of pointing device.

NSFont.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAntialiasThresholdChangedNotification	Posted after the threshold for anti-aliasing changes.
NSFontAntialiasedIntegerAdvancementsRenderingMode	Specifies antialiased, integer advancements rendering mode.
NSFontAntialiasedRenderingMode	Specifies antialiased, floating-point advancements rendering mode (synonymous with printerFont).
NSFontDefaultRenderingMode	Determines the actual mode based on the user preference settings.
NSFontIntegerAdvancementsRenderingMode	Specifies integer advancements rendering mode.
NSFontRenderingMode	These constants specify the font rendering mode.

NSFontSetChangedNotification	Posted after the the currently-set font changes.

NSFontDescriptor.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSFontBoldTrait	The font's typestyle is boldface.
NSFontCascadeListAttribute	An NSArray instance—each member of the array is a sub-descriptor.
NSFontCharacterSetAttribute	An NSCharacterSet instance that represents the set of Unicode characters covered by the font.
NSFontClarendonSerifsClass	The font's style is a variation of the Oldstyle Serifs and the Transitional Serifs.
NSFontCondensedTrait	The font's typestyle is condensed. Expanded and condensed traits are mutually exclusive.
NSFontExpandedTrait	The font's typestyle is expanded. Expanded and condensed traits are mutually exclusive.
NSFontFamilyClass	These constants classify certain stylistic qualities of the font. These values correspond closely to the font class values in the OpenType OS/2 table. The class values are bundled in the upper four bits of the NSFontSymbolicTraits and can be accessed via NSFontFamilyClassMask. For more information about the specific meaning of each identifier, refer to the OpenType specification.
NSFontFamilyClassMask	The font family class mask used to access NSFontFamilyClass values.
NSFontFixedAdvanceAttribute	An NSNumber instance containing a float value that overrides the glyph advancement specified by the font.
NSFontFreeformSerifsClass	The font's style includes serifs, but it expresses a design freedom that does not generally fit within the other serif design classifications.
NSFontItalicTrait	The font's typestyle is italic.
NSFontMatrixAttribute	An NSAffineTransform instance that specifies the font's transformation matrix.

`NSFontModernSerifsClass`	The font's style is based on the Latin printing style of the 20th century.
`NSFontMonoSpaceTrait`	The font uses fixed-pitch glyphs if available. The font may have multiple glyph advances (many CJK glyphs contain two spaces).
`NSFontOldStyleSerifsClass`	The font's style is based on the Latin printing style of the 15th to 17th century.
`NSFontOrnamentalsClass`	The font's style includes highly decorated or stylized character shapes such as those typically used in headlines.
`NSFontSansSerifClass`	The font's style includes most basic letter forms (excluding Scripts and Ornamentals) that do not have serifs on the strokes.
`NSFontScriptsClass`	The font's style is among those typefaces designed to simulate handwriting.
`NSFontSlabSerifsClass`	The font's style is characterized by serifs with a square transition between the strokes and the serifs (no brackets).
`NSFontSlantTrait`	The relative slant angle value as an NSNumber object.
`NSFontSymbolicClass`	The font's style is generally design independent, making it suitable for special characters (icons, dingbats, technical symbols, and so on) that may be used equally well with any font.
`NSFontSymbolicTrait`	The symbolic traits value as an NSNumber object.
`NSFontSymbolicTraits`	NSFontSymbolicTraits symbolically describes stylistic aspects of a font.
`NSFontTraitsAttribute`	An NSDictionary instance instance fully describing font traits.
`NSFontTransitionalSerifsClass`	The font's style is based on the Latin printing style of the 18th to 19th century.
`NSFontUIOptimizedTrait`	The font synthesizes appropriate attributes for user interface rendering, such as control titles, if necessary.
`NSFontUnknownClass`	The font has no design classification.
`NSFontVariationAttribute`	An NSDictionary instance that describes the font's variation axis.
`NSFontVariationAxisDefaultValueKey`	The default axis value as an NSNumber object.
`NSFontVariationAxisIdentifierKey`	The axis identifier value as an NSNumber object.
`NSFontVariationAxisMaximumValueKey`	The maximum axis value as an NSNumber object.
`NSFontVariationAxisMinimumValueKey`	The minimum axis value as an NSNumber object.

NSFontVariationAxisNameKey	The localized variation axis name.
NSFontVerticalTrait	The font uses vertical glyph variants and metrics.
NSFontWeightTrait	The normalized weight value as an NSNumber object.
NSFontWidthTrait	The relative inter-glyph spacing value as an NSNumber object.

NSFontPanel.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSFontPanelAllEffectsModeMask	Display all the effects user interface items.
NSFontPanelDocumentColorEffectModeMask	Display the document color button.
NSFontPanelShadowEffectModeMask	Display the shadow effects button.
NSFontPanelStrikethroughEffectModeMask	Display the strike-through popup menu.
NSFontPanelTextColorEffectModeMask	Display the text color button.
NSFontPanelUnderlineEffectModeMask	Display the underline popup menu.

NSKeyValueBinding.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAlignmentBinding	See Cocoa Bindings Reference for more information.
NSAllowsEditingMultipleValuesSelectionBindingOption	An NSNumber object containing a Boolean value that determines if the binding allows editing when the value represents a multiple selection.
NSAllowsNullArgumentBindingOption	An NSNumber object containing a Boolean value that determines if the argument bindings allows passing argument values of nil.

NSAlternateImageBinding	See Cocoa Bindings Reference for more information.
NSAlternateTitleBinding	See Cocoa Bindings Reference for more information.
NSAlwaysPresentsApplicationModalAlertsBindingOption	
NSAnimateBinding	See Cocoa Bindings Reference for more information.
NSAnimationDelayBinding	See Cocoa Bindings Reference for more information.
NSArgumentBinding	See Cocoa Bindings Reference for more information.
NSAttributedStringBinding	See Cocoa Bindings Reference for more information.
NSConditionallySetsEditableBindingOption	An NSNumber object containing a Boolean value that determines if the editable state of the user interface item is automatically configured based on the controller's selection.
NSConditionallySetsEnabledBindingOption	An NSNumber object containing a Boolean value that determines if the enabled state of the user interface item is automatically configured based on the controller's selection.
NSConditionallySetsHiddenBindingOption	An NSNumber object containing a Boolean value that determines if the hidden state of the user interface item is automatically configured based on the controller's selection.
NSContentArrayBinding	See Cocoa Bindings Reference for more information.
NSContentArrayForMultipleSelectionBinding	See Cocoa Bindings Reference for more information.
NSContentBinding	See Cocoa Bindings Reference for more information.
NSContentHeightBinding	See Cocoa Bindings Reference for more information.
NSContentObjectBinding	See Cocoa Bindings Reference for more information.

`NSContentObjectsBinding`	See Cocoa Bindings Reference for more information.
`NSContentSetBinding`	See Cocoa Bindings Reference for more information.
`NSContentValuesBinding`	See Cocoa Bindings Reference for more information.
`NSContentWidthBinding`	See Cocoa Bindings Reference for more information.
`NSContinuouslyUpdatesValueBindingOption`	An NSNumber object containing a Boolean value that determines whether the value of the binding is updated as edits are made to the user interface item or is updated only when the user interface item resigns as the responder.
`NSCreatesSortDescriptorBindingOption`	An NSNumber object containing a Boolean value that determines if a sort descriptor is created for a table column.
`NSCriticalValueBinding`	See Cocoa Bindings Reference for more information.
`NSDataBinding`	See Cocoa Bindings Reference for more information.
`NSDeletesObjectsOnRemoveBindingsOption`	An NSNumber object containing a Boolean value that determines if an object is deleted from the managed context immediately upon being removed from a relationship.
`NSDisplayNameBindingOption`	An NSString object containing a human readable string to be displayed for a predicate.
`NSDisplayPatternBindingOption`	An NSString object that specifies a format string used to construct the final value of a string.
`NSDisplayPatternTitleBinding`	See Cocoa Bindings Reference for more information.
`NSDisplayPatternValueBinding`	See Cocoa Bindings Reference for more information.
`NSDocumentEditedBinding`	See Cocoa Bindings Reference for more information.
`NSDoubleClickArgumentBinding`	
`NSDoubleClickTargetBinding`	

NSEditableBinding	See Cocoa Bindings Reference for more information.
NSEnabledBinding	See Cocoa Bindings Reference for more information.
NSFilterPredicateBinding	
NSFontBinding	See Cocoa Bindings Reference for more information.
NSFontBoldBinding	See Cocoa Bindings Reference for more information.
NSFontFamilyNameBinding	See Cocoa Bindings Reference for more information.
NSFontItalicBinding	See Cocoa Bindings Reference for more information.
NSFontNameBinding	See Cocoa Bindings Reference for more information.
NSFontSizeBinding	See Cocoa Bindings Reference for more information.
NSHandlesContentAsCompoundValueBindingOption	An NSNumber object containing a Boolean value that determines if the content is treated as a compound value.
NSHeaderTitleBinding	See Cocoa Bindings Reference for more information.
NSHiddenBinding	See Cocoa Bindings Reference for more information.
NSImageBinding	See Cocoa Bindings Reference for more information.
NSInsertsNullPlaceholderBindingOption	An NSNumber object containing a Boolean value that determines if an additional item which represents nil is inserted into a matrix or pop-up menu before the items in the content array.
NSInvokesSeparatelyWithArrayObjectsBindingOption	An NSNumber object containing a Boolean value that determines whether the specified selector is invoked with the array as the argument or is invoked repeatedly with each array item as an argument.
NSIsIndeterminateBinding	See Cocoa Bindings Reference for more information.

`NSLabelBinding`	See Cocoa Bindings Reference for more information.
`NSManagedObjectContextBinding`	See Cocoa Bindings Reference for more information.
`NSMaximumRecentsBinding`	
`NSMaxValueBinding`	See Cocoa Bindings Reference for more information.
`NSMaxWidthBinding`	See Cocoa Bindings Reference for more information.
`NSMinValueBinding`	See Cocoa Bindings Reference for more information.
`NSMinWidthBinding`	See Cocoa Bindings Reference for more information.
`NSMixedStateImageBinding`	See Cocoa Bindings Reference for more information.
`NSMultipleValuesPlaceholderBindingOption`	An object that is used as a placeholder when the key path of the bound controller returns the NSMultipleValuesMarker marker for a binding.
`NSNoSelectionPlaceholderBindingOption`	An object that is used as a placeholder when the key path of the bound controller returns the NSNoSelectionMarker marker for a binding.
`NSNotApplicablePlaceholderBindingOption`	An object that is used as a placeholder when the key path of the bound controller returns the NSNotApplicableMarker marker for a binding.
`NSNullPlaceholderBindingOption`	An object that is used as a placeholder when the key path of the bound controller returns nil for a binding.
`NSObservedKeyPathKey`	An NSString object containing the key path of the binding.
`NSObservedObjectKey`	The object that is the observable controller of the binding.
`NSOffStateImageBinding`	See Cocoa Bindings Reference for more information.

NSOnStateImageBinding	See Cocoa Bindings Reference for more information.
NSOptionsKey	An NSDictionary object containing key value pairs as specified in the options dictionary when the binding was created.
NSPredicateBinding	See Cocoa Bindings Reference for more information.
NSPredicateFormatBindingOption	An NSString object containing the predicate pattern string for the predicate bindings. Use $value to refer to the value in the search field.
NSRaisesForNotApplicableKeysBindingOption	An NSNumber object containing a Boolean value that specifies if an exception is raised when the binding is bound to a key that is not applicable—for example when an object is not key-value coding compliant for a key.
NSRecentSearchesBinding	See Cocoa Bindings Reference for more information.
NSRepresentedFilenameBinding	See Cocoa Bindings Reference for more information.
NSRowHeightBinding	See Cocoa Bindings Reference for more information.
NSSelectedIdentifierBinding	See Cocoa Bindings Reference for more information.
NSSelectedIndexBinding	See Cocoa Bindings Reference for more information.
NSSelectedLabelBinding	See Cocoa Bindings Reference for more information.
NSSelectedObjectBinding	See Cocoa Bindings Reference for more information.
NSSelectedObjectsBinding	See Cocoa Bindings Reference for more information.
NSSelectedTagBinding	See Cocoa Bindings Reference for more information.
NSSelectedValueBinding	See Cocoa Bindings Reference for more information.
NSSelectedValuesBinding	See Cocoa Bindings Reference for more information.

`NSSelectionIndexesBinding`	See Cocoa Bindings Reference for more information.
`NSSelectionIndexPathsBinding`	See Cocoa Bindings Reference for more information.
`NSSelectorNameBindingOption`	An NSString object that specifies the method selector invoked by the target binding when the user interface item is clicked.
`NSSelectsAllWhenSettingContentBindingOption`	An NSNumber object containing a Boolean value that specifies if all the items in the array controller are selected when the content is set.
`NSSortDescriptorsBinding`	See Cocoa Bindings Reference for more information.
`NSTargetBinding`	See Cocoa Bindings Reference for more information.
`NSTextColorBinding`	See Cocoa Bindings Reference for more information.
`NSTitleBinding`	See Cocoa Bindings Reference for more information.
`NSToolTipBinding`	See Cocoa Bindings Reference for more information.
`NSValidatesImmediatelyBindingOption`	An NSNumber object containing a Boolean value that determines if the contents of the binding are validated immediately.
`NSValueBinding`	See Cocoa Bindings Reference for more information.
`NSValuePathBinding`	See Cocoa Bindings Reference for more information.
`NSValueTransformerBindingOption`	An NSValueTransformer instance that is applied to the bound value.
`NSValueTransformerNameBindingOption`	The value for this key is an identifier of a registered NSValueTransformer instance that is applied to the bound value.
`NSValueURLBinding`	See Cocoa Bindings Reference for more information.
`NSVisibleBinding`	See Cocoa Bindings Reference for more information.

NSWarningValueBinding	See Cocoa Bindings Reference for more information.
NSWidthBinding	See Cocoa Bindings Reference for more information.

NSLayoutManager.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSTypesetterBehavior_10_4	The typesetter behavior introduced in Mac OS X version 10.4.

NSLevelIndicatorCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSContinuousCapacityLevelIndicatorStyle	A style that is often used to indicate conditions such as how much data is on a hard disk.
NSDiscreteCapacityLevelIndicatorStyle	A style similar to audio level indicators in iTunes.
NSLevelIndicatorStyle	
NSRatingLevelIndicatorStyle	A style similar to the star ranking displays provided in iTunes and iPhoto.
NSRelevancyLevelIndicatorStyle	A style that is used to indicate the relevancy of a search result, for example in Mail.

NSOpenGL.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSOpenGLPFAColorFloat	A Boolean attribute.

`NSOpenGLPFAMultisample`	A Boolean attribute.
`NSOpenGLPFASampleAlpha`	A Boolean attribute. If present and used with NSOpenGLPFASampleBuffers and NSOpenGLPFASampleBuffers, this attribute hints to OpenGL to update multi-sample alpha values to ensure the most accurate rendering. If pixel format is not requesting anti-aliasing then this hint does nothing.
`NSOpenGLPFASupersample`	A Boolean attribute.

NSPasteboard.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSInkTextPboardType`	Ink text data.

NSPrintInfo.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSPrintDetailedErrorReporting`	An NSNumber object containing a Boolean value—if YES, produce detailed reports when an error occurs.
`NSPrintFaxNumber`	An NSString object that specifies a fax number.
`NSPrintHeaderAndFooter`	An NSNumber object containing a Boolean value—if YES, a standard header and footer are added outside the margins of each page.
`NSPrintPagesAcross`	An NSNumber object that specifies the number of logical pages to be tiled horizontally on a physical sheet of paper.
`NSPrintPagesDown`	An NSNumber object that specifies the number of logical pages to be tiled vertically on a physical sheet of paper.
`NSPrintPrinterName`	An NSString object that specifies the name of a printer.
`NSPrintTime`	An NSDate object that specifies the time at which printing should begin.

NSStringDrawing.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSStringDrawingDisableScreenFontSubstitution`	Disable screen font substitution (equivalent to [NSLayoutManager setUsesScreenFonts:NO]).
`NSStringDrawingOneShot`	Suppresses caching layout information.
`NSStringDrawingOptions`	The following constants are provided as rendering options for a string when it is drawn.
`NSStringDrawingUsesDeviceMetrics`	Uses image glyph bounds instead of typographic bounds.
`NSStringDrawingUsesFontLeading`	Uses the font leading for calculating line heights.
`NSStringDrawingUsesLineFragmentOrigin`	The specified origin is the line fragment origin, not the baseline origin.

NSTableColumn.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSTableColumnAutoresizingMask`	Allows the table column to resize automatically in response to resizing the tableview. Enabling this option is the same as enabling the "Live Resizable" option in Interface Builder. The resizing behavior for the table view is set using the NSTableView method setColumnAutoresizingStyle:.
`NSTableColumnNoResizing`	Prevents the table column from resizing.
`NSTableColumnUserResizingMask`	Allows the table column to be resized explicitly by the user. Enabling this option is the same as enabling the "User Resizable" option in Interface Builder.

NSTableView.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSTableViewColumnAutoresizingStyle`	The following constants specify the autoresizing styles. These constants are used by columnAutoresizingStyle and setColumnAutoresizingStyle:.
`NSTableViewFirstColumnOnlyAutoresizingStyle`	Autoresize only the first table column.
`NSTableViewLastColumnOnlyAutoresizingStyle`	Autoresize only the last table column.
`NSTableViewNoColumnAutoresizing`	Disable table column autoresizing.
`NSTableViewReverseSequentialColumnAutoresizingStyle`	Autoresize each table column sequentially, from the first auto-resizable column to the last auto-resizable column; proceed to the next column when the current column has reached its minimum or maximum size.
`NSTableViewSequentialColumnAutoresizingStyle`	Autoresize each table column sequentially, from the last auto-resizable column to the first auto-resizable column; proceed to the next column when the current column has reached its minimum or maximum size.
`NSTableViewUniformColumnAutoresizingStyle`	Autoresize all columns by distributing space equally, simultaneously.

NSText.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSWritingDirectionNatural`	The writing direction is determined using the Unicode Bidi Algorithm rules P2 and P3. Default.

NSTextList.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSTextListPrependEnclosingMarker	Specifies that a nested list should include the marker for its enclosing superlist before its own marker.

NSTextTable.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSTextBlockAbsoluteValueType	Absolute value in points
NSTextBlockBaselineAlignment	Aligns adjacent blocks at the baseline of the first line of text in the block
NSTextBlockBorder	Border of the text block: space between padding and margin, typically colored to present a visible boundary
NSTextBlockBottomAlignment	Aligns adjacent blocks at their bottom
NSTextBlockDimension	
NSTextBlockHeight	Height of the text block
NSTextBlockLayer	
NSTextBlockMargin	Margin of the text block: space surrounding the border
NSTextBlockMaximumHeight	Maximum height of the text block
NSTextBlockMaximumWidth	Maximum width of the text block
NSTextBlockMiddleAlignment	Aligns adjacent blocks at their middle
NSTextBlockMinimumHeight	Minimum height of the text block
NSTextBlockMinimumWidth	Minimum width of the text block
NSTextBlockPadding	Padding of the text block: space surrounding the content area extending to the border
NSTextBlockPercentageValueType	Percentage value (out of 100)

NSTextBlockTopAlignment	Aligns adjacent blocks at their top
NSTextBlockValueType	
NSTextBlockVerticalAlignment	
NSTextBlockWidth	Width of the text block
NSTextTableAutomaticLayoutAlgorithm	Specifies automatic layout algorithm
NSTextTableFixedLayoutAlgorithm	Specifies fixed layout algorithm
NSTextTableLayoutAlgorithm	These constants, specifying the type of text table layout algorithm, are used with setLayoutAlgorithm:.

NSTextView.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSFindPanelActionSelectAll	Selects all query instances in the text view.
NSFindPanelActionSelectAllInSelection	Selects all query instances within the selection.

NSTokenFieldCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSDefaultTokenStyle	Style best used for keyword type tokens.
NSPlainTextTokenStyle	Style to use for data you want represented as plain-text and without any token background.
NSRoundedTokenStyle	Style best used for address type tokens.
NSTokenStyle	The NSTokenStyle type provides storage for values defined in “NSTokenStyle Values”.

NSToolbarItem.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSToolbarItemVisibilityPriorityHigh`	Items with this priority are less inclined to be pushed to the overflow menu.
`NSToolbarItemVisibilityPriorityLow`	Items with this priority will be the first items to be pushed to the overflow menu.
`NSToolbarItemVisibilityPriorityStandard`	The default visibility priority.
`NSToolbarItemVisibilityPriorityUser`	Items with this priority are the last to be pushed to the overflow menu. Only the user should set items to this priority.

NSTypesetter.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSTypesetterContainerBreakAction`	Causes container break.
`NSTypesetterControlCharacterAction`	The following constants are possible values returned by the actionForControlCharacterAtIndex: method to determine the action associated with a control character.
`NSTypesetterHorizontalTabAction`	Treated as tab character.
`NSTypesetterLineBreakAction`	Causes line break.
`NSTypesetterParagraphBreakAction`	Causes paragraph break; the value returned by firstLineHeadIndent is the advancement used for the following glyph.
`NSTypesetterWhitespaceAction`	The width for glyphs with this action are determined by boundingBoxForControlGlyphAtIndex: forTextContainer: proposedLineFragment:glyphPosition: characterIndex:, if the method is implemented; otherwise, same as NSTypesetterZeroAdvancementAction.
`NSTypesetterZeroAdvancementAction`	Glyphs with this action are flitered out from layout (notShownAttribute == YES).

NSWindow.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSUnifiedTitleAndToolbarWindowMask`	Specifies a window whose toolbar and title bar are rendered on a single continuous background.
`NSUnscaledWindowMask`	Specifies that the window is created without any scaling factors applied.
`NSWindowDidChangeScreenProfileNotification`	Posted whenever the display profile for the screen containing the window changes.

NSWorkspace.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSExclude10_4ElementsIconCreationOption`	Suppress generation of the new higher resolution icon representations that are supported in Mac OS X v10.4.
`NSExcludeQuickDrawElementsIconCreationOption`	Suppress generation of the QuickDraw format icon representations that are used Mac OS X v10.0 through v10.4.
`NSWorkspaceIconCreationOptions`	

10.3 Symbol Changes

This article lists the symbols added to AppKit.framework in Mac OS X v10.3.

Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

NSAlert (New)

Complete reference information is available in the NSAlert reference.

Class Methods

alertWithMessageText:defaultButton: alternateButton:otherButton: informativeTextWithFormat:	Creates an alert compatible with alerts created using the NSRunAlertPanel function for display as a warning-style alert.

Instance Methods

addButtonWithTitle:	Adds a button with a given title to the receiver.
alertStyle	Returns the NSAlertStyle constant identifying the receiver's alert style.
beginSheetModalForWindow:modalDelegate: didEndSelector:contextInfo:	Runs the receiver modally as an alert sheet attached to a specified window.
buttons	Returns the receiver's buttons.
delegate	Returns the receiver's delegate.
helpAnchor	Returns the receiver's HTML help anchor.
icon	Returns the icon displayed in the receiver.
informativeText	Returns the receiver's informative text.
messageText	Returns the receiver's message text (or title).

runModal	Runs the receiver as an application-modal dialog and returns the constant positionally identifying the button clicked.
setAlertStyle:	Sets the alert style of the receiver.
setDelegate:	Sets the receiver's delegate.
setHelpAnchor:	Associates the receiver to a given anchor.
setIcon:	Sets the icon to be displayed in the alert to a given icon.
setInformativeText:	Sets the receiver's informative text to a given text.
setMessageText:	Sets the receiver's message text, or title, to a given text.
setShowsHelp:	Specifies whether the receiver has a help button.
showsHelp	Indicates whether the receiver has a help button.
window	Provides the application-modal panel associated with the receiver.

Delegate Methods

alertShowHelp:	Sent to the delegate when the user clicks the alert's help button. The delegate causes help to be displayed for an alert, directly or indirectly.

NSApplication

Complete reference information is available in the NSApplication reference.

Instance Methods

orderFrontCharacterPalette:	Opens the character palette.
replyToOpenOrPrint:	Handles errors that might occur when the user attempts to open or print files.

Delegate Methods

application:openFiles:	Tells the delegate to open multiple files.
application:printFiles:	Identical to application:printFile: except that the receiver prints multiple files corresponding to the file names in the filenames array.

NSArrayController (New)

Complete reference information is available in the `NSArrayController` reference.

Instance Methods

`addObject:`	Adds object to the receiver’s content collection and the arranged objects array.
`addObjects:`	Adds objects to the receiver’s content collection.
`addSelectedObjects:`	Adds objects from the receiver’s content array to the current selection, returning YES if the selection was changed.
`addSelectionIndexes:`	Adds the objects at the specified indexes in the receiver’s content array to the current selection, returning YES if the selection was changed.
`arrangedObjects`	Returns an array containing the receiver’s content objects arranged using arrangeObjects:.
`arrangeObjects:`	Returns a given array, appropriately sorted and filtered.
`avoidsEmptySelection`	Returns a Boolean value that indicates whether if the receiver requires that the content array attempt to maintain a selection.
`canInsert`	Returns a Boolean value that indicates whether an object can be inserted into the receiver’s content collection.
`canSelectNext`	Returns YES if the next object, relative to the current selection, in the receiver’s content array can be selected.
`canSelectPrevious`	Returns YES if the previous object, relative to the current selection, in the receiver’s content array can be selected.
`insert:`	Creates a new object and inserts it into the receiver’s content array.
`insertObject:atArrangedObjectIndex:`	Inserts object into the receiver’s arranged objects array at the location specified by index, and adds it to the receiver’s content collection.
`insertObjects:atArrangedObjectIndexes:`	Inserts objects into the receiver’s arranged objects array at the locations specified in indexes, and adds it to the receiver’s content collection.

`preservesSelection`	Returns a Boolean value that indicates whether the receiver will attempt to preserve the current selection then when the content changes.
`rearrangeObjects`	Triggers filtering of the receiver's content.
`removeObject:`	Removes object from the receiver's content collection.
`removeObjectAtArrangedObjectIndex:`	Removes the object at the specified index in the receiver's arranged objects from the receiver's content array.
`removeObjects:`	Removes objects from the receiver's content collection.
`removeObjectsAtArrangedObjectIndexes:`	Removes the objects at the specified indexes in the receiver's arranged objects from the content array.
`removeSelectedObjects:`	Removes objects from the receiver's current selection, returning YES if the selection was changed.
`removeSelectionIndexes:`	Removes the object as the specified indexes from the receiver's current selection, returning YES if the selection was changed.
`selectedObjects`	Returns an array containing the receiver's selected objects.
`selectionIndex`	Returns the index of the first object in the receiver's selection.
`selectionIndexes`	Returns an index set containing the indexes of the receiver's currently selected objects in the content array.
`selectNext:`	Selects the next object, relative to the current selection, in the receiver's arranged content.
`selectPrevious:`	Selects the previous object, relative to the current selection, in the receiver's arranged content.
`selectsInsertedObjects`	Returns whether the receiver automatically selects inserted objects.
`setAvoidsEmptySelection:`	Sets whether the receiver attempts to avoid an empty selection.
`setPreservesSelection:`	Sets whether the receiver attempts to preserve selection when the content changes.
`setSelectedObjects:`	Sets objects as the receiver's current selection, returning YES if the selection was changed.
`setSelectionIndex:`	Sets the receiver's selection to the given index, and returns a Boolean value that indicates whether the selection was changed.

`setSelectionIndexes:`	Sets the receiver’s selection indexes and returns a Boolean value that indicates whether the selection changed.
`setSelectsInsertedObjects:`	Sets whether the receiver will automatically select objects as they are inserted.
`setSortDescriptors:`	Sets the sort descriptors for the receiver.
`sortDescriptors`	Returns the receiver's array of sort descriptors.

NSATSTypesetter (New)

Complete reference information is available in the `NSATSTypesetter` reference.

Class Methods

`sharedTypesetter`	Returns a shared instance of NSATSTypesetter.

Instance Methods

`attributedString`	Returns the text backing store, usually an instance of NSTextStorage.
`bidiProcessingEnabled`	Returns a Boolean value indicating the bidirectional text processing setting currently in effect.

`boundingBoxForControlGlyphAtIndex:`
`forTextContainer:proposedLineFragment:`
`glyphPosition:characterIndex:`
	Returns the bounding rectangle for the given control glyph, at the given glyph position and character index, in the given text container.

`characterRangeForGlyphRange:actualGlyphRange:`	Returns the range for the characters in the receiver’s text store that are mapped to the glyphs in the given glyph range.
`currentTextContainer`	Returns the text container for the text being typeset.
`deleteGlyphsInRange:`	Deletes the glyphs in the given glyph range from the glyph cache maintained by the layout manager.

`getGlyphsInRange:glyphs:characterIndexes:`
`glyphInscriptions:elasticBits:`
	Extracts the information needed to lay out the glyphs in the given glyph buffer from the given glyph range.

getLineFragmentRect:usedRect: forParagraphSeparatorGlyphRange:atProposedOrigin:	Calculates the line fragment rectangle and line fragment used rectangle for blank lines.
glyphRangeForCharacterRange:actualCharacterRange:	Returns the range for the glyphs mapped to the characters of the text store in the given character range.
hyphenationFactor	Returns the current hyphenation factor.
hyphenationFactorForGlyphAtIndex:	Returns the hyphenation factor in effect at the given glyph index.
hyphenCharacterForGlyphAtIndex:	Returns the hyphen character to be inserted after the given glyph when hyphenation is enabled in the layout manager.
insertGlyph:atGlyphIndex:characterIndex:	Enables the typesetter to insert a new glyph into the stream.
layoutManager	Returns the layout manager for the text being typeset.
layoutParagraphAtPoint:	Lays out glyphs in the current glyph range until the next paragraph separator is reached.
lineFragmentPadding	Returns the current line fragment padding amount; that is, the portion on each end of the line fragment rectangle left blank.
lineFragmentRectForProposedRect:remainingRect:	This method has been deprecated. Use the NSTypesetter method getLineFragmentRect:usedRect: remainingRect: forStartingGlyphAtIndex: proposedRect:lineSpacing: paragraphSpacingBefore: paragraphSpacingAfter: instead.
lineSpacingAfterGlyphAtIndex: withProposedLineFragmentRect:	Returns the line spacing in effect following the specified glyph.
paragraphGlyphRange	Returns the glyph range currently being processed.
paragraphSeparatorGlyphRange	Returns the current paragraph separator range, which is the full range that contains the current glyph range and that extends from one paragraph separator character to the next.

`paragraphSpacingAfterGlyphAtIndex:`
`withProposedLineFragmentRect:`
	Returns the paragraph spacing, the number of points of space added following a paragraph, which is in effect after the specified glyph.

`paragraphSpacingBeforeGlyphAtIndex:`
`withProposedLineFragmentRect:`
	Returns the number of points of space added before a paragraph, which is in effect before the specified glyph.

`setAttachmentSize:forGlyphRange:`	Sets the size the glyphs in the given glyph range (assumed to be attachments) will be asked to draw themselves.
`setAttributedString:`	Sets the text backing store on which this typesetter operates.
`setBidiLevels:forGlyphRange:`	Sets the direction of the glyphs in the given glyph range for bidirectional text to the given levels.
`setBidiProcessingEnabled:`	Sets a Boolean value controlling whether the typesetter performs bidirectional text processing.
`setDrawsOutsideLineFragment:forGlyphRange:`	Sets a Boolean value controlling whether the glyphs in the given glyph range exceed the bounds of the line fragment in which they are laid out.
`setHardInvalidation:forGlyphRange:`	Sets a Boolean value controlling whether to force the layout manager to invalidate the portion of the glyph cache in the given glyph range when invalidating layout.
`setHyphenationFactor:`	Sets the threshold controlling when hyphenation is attempted
`setLineFragmentPadding:`	Sets the amount (in points) by which text is inset within line fragment rectangles

`setLineFragmentRect:forGlyphRange:`
`usedRect:baselineOffset:`
	Sets the line fragment rectangle where the glyphs in in the given glyph range are laid out to the given line fragment rectangle.

`setLocation:withAdvancements:forStartOfGlyphRange:`	Sets the location where the glyphs in the given glyph range are laid out to the specified location.

setNotShownAttribute:forGlyphRange:	Sets a Boolean value controlling whether the glyphs in the given glyph rangeare not shown.
setParagraphGlyphRange:separatorGlyphRange:	Sets the current glyph range being processed and the paragraph separator glyph range (the range of the paragraph separator character or characters).
setTypesetterBehavior:	Sets the default typesetter behavior, which affects glyph spacing and line height.
setUsesFontLeading:	Sets a Boolean value controlling whether the typesetter uses the leading (or line gap) value specified in the font metric information.
shouldBreakLineByHyphenatingBeforeCharacterAtIndex:	The typesetter calls this method, if implemented by a subclass, before breaking a line by hyphenating before the character at the given character index, enabling the subclass to control line breaking.
shouldBreakLineByWordBeforeCharacterAtIndex:	The typesetter calls this method, if implemented by a subclass, before breaking a line by word wrapping before the character at the given character index, enabling the subclass to control line breaking.
substituteFontForFont:	Returns a screen font suitable for use in place of the specified original font, or simply returns the original font if a screen font can't be used or isn't available.
substituteGlyphsInRange:withGlyphs:	Replaces the glyphs in the given glyph range with the given glyphs.
textTabForGlyphLocation:writingDirection: maxLocation:	Returns the text tab next closest to a given glyph location, indexing in the specified direction but not beyond a given glyph location.
typesetterBehavior	Returns the current typesetter behavior value.

`usesFontLeading`	Returns a Boolean value indicating whether the typesetter uses the leading (or line gap) value specified in the font metric information of the current font.
`willSetLineFragmentRect:forGlyphRange:usedRect:baselineOffset:`	Called by the typesetter just prior to calling setLineFragmentRect:forGlyphRange:usedRect:baselineOffset: which stores the actual line fragment rectangle location in the layout manager.

NSAttributedString

Complete reference information is available in the `NSAttributedString` reference.

Instance Methods

`docFormatFromRange:documentAttributes:`	Returns an NSData object that contains a Microsoft Word–format stream corresponding to the characters and attributes within the specified range.
`initWithDocFormat:documentAttributes:`	Initializes and returns a new NSAttributedString object from Microsoft Word format data contained in the given NSData object.
`initWithHTML:options:documentAttributes:`	Initializes and returns a new NSAttributedString object from HTML contained in the given data object.
`lineBreakByHyphenatingBeforeIndex:withinRange:`	Returns the index of the closest character before the given index, and within the given range, that can be placed on a new line by hyphenating.

NSBrowser

Complete reference information is available in the `NSBrowser` reference.

Class Methods

`removeSavedColumnsWithAutosaveName:`	Removes the column configuration data stored under the given name from the application's user defaults.

Instance Methods

`columnContentWidthForColumnWidth:`	Given the column width, returns the content width.
`columnResizingType`	Returns the receiver’s column resizing type.
`columnsAutosaveName`	Returns the name used to automatically save the receiver’s column configuration.
`columnWidthForColumnContentWidth:`	Given the content width, returns the column width.
`prefersAllColumnUserResizing`	Returns a Boolean value indicating if the browser is set to resize all columns simultaneously rather than resizing a single column at a time.
`setColumnResizingType:`	Sets the receiver’s column resizing type.
`setColumnsAutosaveName:`	Sets the name used to automatically save the receiver’s column configuration.
`setPrefersAllColumnUserResizing:`	Specifies whether the browser resizes all columns simultaneously rather than resizing a single column at a time.
`setWidth:ofColumn:`	Sets the width of the specified column.
`widthOfColumn:`	Returns the width of the specified column.

Delegate Methods

`browser:shouldSizeColumn:forUserResize:toWidth:`	Used for determining a column’s initial size.
`browser:sizeToFitWidthOfColumn:`	Returns the ideal width for a column.

NSCell

Complete reference information is available in the `NSCell` reference.

Class Methods

`defaultFocusRingType`	Returns the default type of focus ring for the receiver.

Instance Methods

`focusRingType`	Returns the type of focus ring currently set for the receiver.
`setFocusRingType:`	Sets the type of focus ring to be used.

NSColor

Complete reference information is available in the `NSColor` reference.

Class Methods

`controlAlternatingRowBackgroundColors`	Returns an array containing the system specified background colors for alternating rows in tables and lists.
`currentControlTint`	Returns the current system control tint.

Instance Methods

`setFill`	Sets the fill color of subsequent drawing to the receiver's color.
`setStroke`	Sets the stroke color of subsequent drawing to the receiver's color.

NSComboBox

Complete reference information is available in the `NSComboBox` reference.

Instance Methods

`isButtonBordered`	Returns whether the combo box button is set to display a border.
`setButtonBordered:`	Determines whether the button in the combo box is displayed with a border.

NSComboBoxCell

Complete reference information is available in the `NSComboBoxCell` reference.

Instance Methods

`isButtonBordered`	Returns a Boolean value indicating whether the combo box button is set to display a border.
`setButtonBordered:`	Determines whether the button in the combo box is displayed with a border.

NSControl

Complete reference information is available in the NSControl reference.

Delegate Methods

control:textView:completions:forPartialWordRange:indexOfSelectedItem:	Invoked to allow you to control the list of proposed text completions generated by text fields and other controls.

NSController (New)

Complete reference information is available in the NSController reference.

Instance Methods

commitEditing	Causes the receiver to attempt to commit any pending edits, returning YES if successful or no edits were pending.
discardEditing	Discards any pending changes by registered editors.
isEditing	Returns YES if there are any editors currently registered with the receiver, NO otherwise.
objectDidBeginEditing:	Invoked to inform the receiver that editor has uncommitted changes that can affect the receiver.
objectDidEndEditing:	Invoked to inform the receiver that editor has committed or discarded its changes.

NSCursor

Complete reference information is available in the NSCursor reference.

Class Methods

closedHandCursor	Returns the closed-hand system cursor.
crosshairCursor	Returns the cross-hair system cursor.
disappearingItemCursor	Returns a cursor indicating that the current operation will result in a disappearing item.
openHandCursor	Returns the open-hand system cursor.
pointingHandCursor	Returns the pointing-hand system cursor.

`resizeDownCursor`	Returns the resize-down system cursor.
`resizeLeftCursor`	Returns the resize-left system cursor.
`resizeLeftRightCursor`	Returns the resize-left-and-right system cursor.
`resizeRightCursor`	Returns the resize-right system cursor.
`resizeUpCursor`	Returns the resize-up system cursor.
`resizeUpDownCursor`	Returns the resize-up-and-down system cursor.

NSDocument

Complete reference information is available in the `NSDocument` reference.

Instance Methods

`windowForSheet`	Returns the most appropriate window, of the windows associated with the receiver, to use as the parent window of a document-modal sheet.

NSFont

Complete reference information is available in the `NSFont` reference.

Class Methods

`menuBarFontOfSize:`	Returns the font used for menu bar items, in the specified size.
`systemFontSizeForControlSize:`	Returns the font size used for the specified control size.

Instance Methods

`fontDescriptor`	Returns the receiver's font descriptor.

NSFontDescriptor (New)

Complete reference information is available in the `NSFontDescriptor` reference.

Class Methods

`fontDescriptorWithFontAttributes:`	Returns a font descriptor with a dictionary of attributes.

fontDescriptorWithName:size:	Returns a font descriptor with the NSFontNameAttribute and NSFontSizeAttribute dictionary attributes set to the given values.

Instance Methods

fontAttributes	Returns the receiver's dictionary of attributes.
initWithFontAttributes:	Initializes and returns a new font descriptor with the specified attributes.

NSFontManager

Complete reference information is available in the `NSFontManager` reference.

Instance Methods

addCollection:options:	Adds a specified font collection to the font manager with a given set of options.
addFontDescriptors:toCollection:	Adds an array of font descriptors to the specified font collection.
availableFontNamesMatchingFontDescriptor:	Returns the names of the fonts that match the attributes in the given font descriptor.
collectionNames	Returns the names of the currently loaded font collections.
convertAttributes:	Converts attributes in response to an object initiating an attribute change, typically the Font panel or Font menu.
fontDescriptorsInCollection:	Returns an array of the font descriptors in the collection specified by the given collection name.
orderFrontStylesPanel:	This action method opens the Font styles panel.
removeCollection:	Removes the specified font collection.
removeFontDescriptor:fromCollection:	Removes the specified font descriptor from the specified collection.
setSelectedAttributes:isMultiple:	Informs the paragraph and character formatting panels when text in a selection has changed attributes.

NSFontPanel

Complete reference information is available in the `NSFontPanel` reference.

Delegate Methods

`validModesForFontPanel:`	Returns the mode mask corresponding to the expected font panel mode.

NSForm

Complete reference information is available in the `NSForm` reference.

Instance Methods

`setFrameSize:`	Sets the size of the receiver's frame size to the specified value.

NSGlyphGenerator (New)

Complete reference information is available in the `NSGlyphGenerator` reference.

Class Methods

`sharedGlyphGenerator`	Returns a shared instance of NSGlyphGenerator.

Instance Methods

`generateGlyphsForGlyphStorage: desiredNumberOfCharacters:glyphIndex: characterIndex:`	Generates glyphs for the specified glyph storage object (NSLayoutManager by default).

NSHelpManager

Complete reference information is available in the `NSHelpManager` reference.

Instance Methods

`findString:inBook:`	Performs a search for the specified string in the specified book.

`openHelpAnchor:inBook:`	Finds and displays the text at the given anchor location in the given book.

NSImageView

Complete reference information is available in the `NSImageView` reference.

Instance Methods

`animates`	Returns a Boolean value indicating whether the receiver automatically plays animated images.
`setAnimates:`	Sets whether the receiver automatically plays an animated image that is assigned to it.

NSLayoutManager

Complete reference information is available in the `NSLayoutManager` reference.

Instance Methods

`drawStrikethroughForGlyphRange:strikethroughType:baselineOffset:lineFragmentRect:lineFragmentGlyphRange:containerOrigin:`	Draws a strikethrough for the glyphs in a given range.
`strikethroughGlyphRange:strikethroughType:lineFragmentRect:lineFragmentGlyphRange:containerOrigin:`	Calculates and draws strikethrough for the glyphs in the given range.

NSMenu

Complete reference information is available in the `NSMenu` reference.

Class Methods

`popUpContextMenu:withEvent:forView:withFont:`	Displays a contextual menu over a view for an event using a specified font.

Instance Methods

`delegate`	Returns the receiver's delegate.

`setDelegate:`	Sets the receiver's delegate.

Delegate Methods

`menu:updateItem:atIndex:shouldCancel:`	Called to let the delegate update a menu item before it is displayed.
`menuHasKeyEquivalent:forEvent:target:action:`	Called to allow the delegate to return the target and action for a key-down event.
`menuNeedsUpdate:`	Called when a menu is about to be displayed at the start of a tracking session so the delegate can modify the menu.
`numberOfItemsInMenu:`	Called when a menu is about to be displayed at the start of a tracking session so the delegate can specify the number of items in the menu.

NSMenuItem

Complete reference information is available in the `NSMenuItem` reference.

Instance Methods

`attributedTitle`	Returns the custom title string for a menu item.
`indentationLevel`	Returns the menu item indentation level for the receiver.
`isAlternate`	Returns a Boolean value that indicates whether the receiver is an alternate to the previous menu item.
`setAlternate:`	Marks the receiver as an alternate to the previous menu item.
`setAttributedTitle:`	Specifies a custom string for a menu item.
`setIndentationLevel:`	Sets the menu item indentation level for the receiver.
`setToolTip:`	Sets a help tag for a menu item.
`toolTip`	Returns the help tag for a menu item.

NSMovieView

Complete reference information is available in the `NSMovieView` reference.

Instance Methods

delete:	This action method deletes the current movie selection from the movie, placing it on the clipboard.

NSMutableAttributedString

Complete reference information is available in the NSMutableAttributedString reference.

Instance Methods

readFromData:options:documentAttributes:	Sets the contents of the receiver from the stream at data.

NSMutableParagraphStyle

Complete reference information is available in the NSMutableParagraphStyle reference.

Instance Methods

setDefaultTabInterval:	Sets the default tab interval for the receiver.
setLineHeightMultiple:	Sets the line height multiple for the receiver.
setParagraphSpacingBefore:	Sets the distance between the paragraph’s top and the beginning of its text content

NSNib (New)

Complete reference information is available in the NSNib reference.

Instance Methods

initWithContentsOfURL:	Returns an NSNib object initialized to the nib file at the specified URL.
initWithNibNamed:bundle:	Returns an NSNib object initialized to the nib file in the specified bundle.
instantiateNibWithExternalNameTable:	Unarchives and instantiates the in-memory contents of the receiver's nib file, creating a distinct object tree and top level objects.

instantiateNibWithOwner:topLevelObjects:	Unarchives and instantiates the in-memory contents of the receiver's nib file, creating a distinct object tree and set of top level objects.

NSObject

Complete reference information is available in the NSObject reference.

Class Methods

defaultPlaceholderForMarker:withBinding:	Returns an object that will be used as the placeholder for the binding, when a key value coding compliant property of an instance of the receiving class returns the value specified by marker, and no other placeholder has been specified.
exposeBinding:	Exposes the specified binding, advertising its availability.
setDefaultPlaceholder:forMarker:withBinding:	Sets placeholder as the default placeholder for the binding, when a key value coding compliant property of an instance of the receiving class returns the value specified by marker, and no other placeholder has been specified.

Instance Methods

accessibilityAttributeValue:forParameter:	Returns the value of the receiver's parameterized attribute corresponding to attribute and parameter. If you implement this method you should also implement accessibilityParameterizedAttributeNames.
accessibilityParameterizedAttributeNames	Returns a list of parameterized attribute names supported by the receiver. If you implement this method you should also implement accessibilityAttributeValue: forParameter:.
bind:toObject:withKeyPath:options:	Establishes a binding between a given property of the receiver and the property of a given object specified by a given key path.
browserColumnConfigurationDidChange:	Used by clients to implement their own column width persistence.
commitEditing	Returns whether the receiver was able to commit any pending edits.

discardEditing	Causes the receiver to discard any changes, restoring the previous values.
exposedBindings	Returns an array containing the bindings exposed by the receiver.
objectDidBeginEditing:	This message should be sent to the receiver when editor has uncommitted changes that can affect the receiver.
objectDidEndEditing:	This message should be sent to the receiver when editor has finished editing a property belonging to the receiver.
panel:directoryDidChange:	Tells the delegate that the user has changed the selected directory in the NSSavePanel object specified.
panelSelectionDidChange:	Tells the delegate that the user changed the selection in the specified Save panel.
textViewDidChangeTypingAttributes:	Sent when a text view's typing attributes change.
unbind:	Removes a given binding between the receiver and a controller.
valueClassForBinding:	Returns the class of the value that will be returned for the specified binding.

NSObjectController (New)

Complete reference information is available in the NSObjectController reference.

Instance Methods

add:	Creates a new object and sets it as the receiver's content object.
addObject:	Sets the receiver's content object.
automaticallyPreparesContent	Returns a Boolean value that indicates whether the receiver automatically prepares its content when it is loaded from a nib.
canAdd	Returns a Boolean value that indicates whether an object can be added to the receiver using add:.
canRemove	Returns a Boolean value that indicates whether an object can be removed from the receiver.
content	Returns the receiver's content object.

initWithContent:	Initializes and returns an NSObjectController object with the given content.
isEditable	Returns a Boolean value that indicates whether the receiver allows adding and removing objects.
newObject	Creates and returns a new object of the appropriate class.
objectClass	Returns the class used when creating new non-Core Data objects.
prepareContent	Typically overridden by subclasses that require additional control over the creation of new objects.
remove:	Removes the receiver's content object.
removeObject:	Removes a given object from the receiver's content.
selectedObjects	Returns an array of all objects to be affected by editing.
selection	Returns a proxy object representing the receiver's selection.
setAutomaticallyPreparesContent:	Sets whether the receiver automatically creates and inserts new content objects automatically when loading from a nib file.
setContent:	Sets the receiver's content object.
setEditable:	Sets whether the receiver allows adding and removing objects.
setObjectClass:	Sets the object class to use when creating new objects.
validateMenuItem:	Returns a Boolean value that indicates whether a given menu item should be enabled.

NSOpenGLContext

Complete reference information is available in the NSOpenGLContext reference.

Instance Methods

CGLContextObj	Returns the low-level, platform-specific Core OpenGL (CGL) context object represented by the receiver.
pixelBuffer	Returns the pixel-buffer object attached to the receiver.
pixelBufferCubeMapFace	Returns the cube map face of the pixel buffer attached to the receiver.

pixelBufferMipMapLevel	Returns the mipmap level of the pixel buffer attached to the receiver.
setPixelBuffer:cubeMapFace:mipMapLevel: currentVirtualScreen:	Attaches the specified pixel buffer to the receiver.
setTextureImageToPixelBuffer:colorBuffer:	Attaches the image data in the specified pixel buffer to the texture object currently bound by the receiver.

NSOpenGLPixelBuffer (New)

Complete reference information is available in the NSOpenGLPixelBuffer reference.

Instance Methods

initWithTextureTarget:textureInternalFormat: textureMaxMipMapLevel:pixelsWide:pixelsHigh:	Returns an NSOpenGLPixelBuffer object initialized with the specified parameters.
pixelsHigh	Returns the height of the receiver's texture (in pixels).
pixelsWide	Returns the width of the receiver's texture (in pixels).
textureInternalFormat	Returns the internal format of the receiver's texture.
textureMaxMipMapLevel	Returns the maximum mipmap level of the receiver's texture.
textureTarget	Returns the texture target of the receiver.

NSOpenGLPixelFormat

Complete reference information is available in the NSOpenGLPixelFormat reference.

Instance Methods

CGLPixelFormatObj	Returns the low-level, platform-specific Core OpenGL (CGL) pixel format object represented by the receiver.

NSOpenGLView

Complete reference information is available in the NSOpenGLView reference.

Instance Methods

prepareOpenGL	Used by subclasses to initialize OpenGL state.

NSOpenPanel

Complete reference information is available in the NSOpenPanel reference.

Instance Methods

beginForDirectory:file:types:modelessDelegate: didEndSelector:contextInfo:	Presents a modeless Open panel.

NSOutlineView

Complete reference information is available in the NSOutlineView reference.

Delegate Methods

outlineView:didClickTableColumn:	Sent at the time the mouse button subsequently goes up in outlineView and tableColumn has been "clicked" without having been dragged anywhere.
outlineView:didDragTableColumn:	Sent at the time the mouse button goes up in outlineView and tableColumn has been dragged during the time the mouse button was down.
outlineView:mouseDownInHeaderOfTableColumn:	Sent to the delegate whenever the mouse button is clicked in outlineView while the cursor is in a column header tableColumn.
outlineView:sortDescriptorsDidChange:	Invoked by an outline view to notify the data source that the descriptors changed and the data may need to be resorted.

NSParagraphStyle

Complete reference information is available in the NSParagraphStyle reference.

Instance Methods

defaultTabInterval	Returns the document-wide default tab interval.

lineHeightMultiple	Returns the line height multiple.
paragraphSpacingBefore	Returns the distance between the paragraph's top and the beginning of its text content.

NSPopUpButtonCell

Complete reference information is available in the NSPopUpButtonCell reference.

Instance Methods

objectValue	Returns the index of the selected item.
setObjectValue:	Selects the item at a specific index using an object value.

NSPrintInfo

Complete reference information is available in the NSPrintInfo reference.

Instance Methods

localizedPaperName	Returns the human-readable name of the currently selected paper size, suitable for presentation in user interfaces.

NSResponder

Complete reference information is available in the NSResponder reference.

Instance Methods

cancelOperation:	Implemented by subclasses to cancel the current operation.
deleteBackwardByDecomposingPreviousCharacter:	Implemented by subclasses to delete the selection, if there is one, or a single character backward from the insertion point.
moveLeftAndModifySelection:	Implemented by subclasses to expand or reduce either end of the selection to the left (display order) by one element or character.
moveRightAndModifySelection:	Implemented by subclasses to expand or reduce either end of the selection to the right (display order) by one element or character.

moveWordLeft:	Implemented by subclasses to move the selection or insertion point one word to the left, in display order.
moveWordLeftAndModifySelection:	Implemented by subclasses to expand or reduce either end of the selection left by one whole word in display order.
moveWordRight:	Implemented by subclasses to move the selection or insertion point one word right.
moveWordRightAndModifySelection:	Implemented by subclasses to expand or reduce either end of the selection to the right by one whole word.

NSSavePanel

Complete reference information is available in the NSSavePanel reference.

Instance Methods

allowedFileTypes	Returns an array of the allowed file types.
allowsOtherFileTypes	Returns a Boolean value that indicates whether the panel allows the user to save files with an extension that's not in the list of allowed types.
canCreateDirectories	Returns a Boolean value that indicates whether the panel allows the user to create directories.
canSelectHiddenExtension	Returns a Boolean value that indicates whether the panel allows the user to hide or show extensions.
delegate	Returns the panel's delegate.
message	Returns the message displayed in the save panel.
nameFieldLabel	Returns the string displayed in front of the filename text field.
setAllowedFileTypes:	Specifies the allowed file types for the panel.
setAllowsOtherFileTypes:	Sets whether the panel allows the user to save files with an extension that's not in the list of allowed types.
setCanCreateDirectories:	Sets whether the panel allows the user to create directories.
setMessage:	Sets the message text displayed in the panel.
setNameFieldLabel:	Sets the text displayed in front of the text field.

NSScrollView

Complete reference information is available in the `NSScrollView` reference.

Instance Methods

`autohidesScrollers`	Returns YES when autohiding is set for scroll bars in the receiver.
`setAutohidesScrollers:`	Determines whether the receiver automatically hides its scroll bars when they are not needed.

NSSearchField (New)

Complete reference information is available in the `NSSearchField` reference.

Instance Methods

`recentsAutosaveName`	Returns the key under which the prior list of recent search strings has been archived.
`recentSearches`	Returns the list of recent search strings for the control.
`setRecentsAutosaveName:`	Sets the autosave name under which the receiver automatically archives the list of recent search strings.
`setRecentSearches:`	Sets the list of recent search strings to list in the pop-up icon menu of the receiver.

NSSearchFieldCell (New)

Complete reference information is available in the `NSSearchFieldCell` reference.

Instance Methods

`cancelButtonCell`	Returns the button cell object used to display the cancel-button image.
`cancelButtonRectForBounds:`	Modifies the bounding rectangle for the cancel button cell.
`maximumRecents`	Returns the maximum number of recent search strings to display in the custom search menu.
`recentsAutosaveName`	Returns the key under which the prior list of recent search strings has been archived.
`recentSearches`	Returns the list of recent search strings for the control.

`resetCancelButtonCell`	Resets the cancel button cell to its default attributes.
`resetSearchButtonCell`	Resets the search button cell to its default attributes.
`searchButtonCell`	Returns the button cell used to display the search-button image.
`searchButtonRectForBounds:`	Modifies the bounding rectangle for the search button cell.
`searchMenuTemplate`	Returns the menu template object used to dynamically construct the search pop-up icon menu.
`searchTextRectForBounds:`	Modifies the bounding rectangle for the search-text field cell.
`sendsWholeSearchString`	Returns a Boolean value indicating whether the receiver sends the search action message when the user clicks the search button (or presses return) or after each keystroke.
`setCancelButtonCell:`	Sets the button cell object used to display the cancel-button image
`setMaximumRecents:`	Sets the maximum number of search strings that can appear in the search menu
`setRecentsAutosaveName:`	Sets the autosave name under which the receiver automatically archives the list of recent search strings.
`setRecentSearches:`	Sets the list of recent search strings to list in the pop-up icon menu of the receiver.
`setSearchButtonCell:`	Sets the button cell used to display the search-button image
`setSearchMenuTemplate:`	Sets the menu template object used to dynamically construct the receiver's pop-up icon menu.
`setSendsWholeSearchString:`	Sets whether the receiver sends the search action message when the user clicks the search button (or presses return) or after each keystroke.

NSSegmentedCell (New)

Complete reference information is available in the `NSSegmentedCell` reference.

Instance Methods

`drawSegment:inFrame:withView:`	Draws the segment in the specified view.
`imageForSegment:`	Returns the image associated with the specified segment.
`isEnabledForSegment:`	Returns a Boolean value indicating whether the specified segment is enabled.
`isSelectedForSegment:`	Returns a Boolean value indicating whether the specified segment is selected,

labelForSegment:	Returns the label of the specified segment.
makeNextSegmentKey	Selects the next segment.
makePreviousSegmentKey	Selects the previous segment.
menuForSegment:	Returns the menu for the specified segment.
segmentCount	Returns the number of segments in the receiver.
selectedSegment	Returns the index of the selected segment of the receiver.
setEnabled:forSegment:	Sets the enabled state of the specified segment
setImage:forSegment:	Sets the image for the specified segment.
setLabel:forSegment:	Sets the label for the specified segment.
setMenu:forSegment:	Sets the menu for the specified segment.
setSegmentCount:	Sets the number of segments in the receiver.
setSelected:forSegment:	Sets the selection state of the specified segment.
setSelectedSegment:	Sets the selected segment of the receiver.
setTag:forSegment:	Sets the tag for the specified segment.
setToolTip:forSegment:	Sets the tool tip for the specified segment.
setTrackingMode:	Sets the tracking mode for the receiver.
setWidth:forSegment:	Sets the width of the specified segment.
tagForSegment:	Returns the tag of the specified segment.
toolTipForSegment:	Returns the tool tip of the specified segment.
trackingMode	Returns the tracking mode of the receiver.
widthForSegment:	Returns the width of the specified segment.

NSSegmentedControl (New)

Complete reference information is available in the `NSSegmentedControl` reference.

Instance Methods

imageForSegment:	Returns the image associated with the specified segment.
isEnabledForSegment:	Returns a Boolean value indicating whether the specified segment is enabled.

isSelectedForSegment:	Returns a Boolean value indicating whether the specified segment is selected.
labelForSegment:	Returns the label of the specified segment
menuForSegment:	Returns the menu for the specified segment
segmentCount	Returns the number of segments in the receiver.
selectedSegment	Returns the index of the selected segment of the receiver.
setEnabled:forSegment:	Sets the enabled state of the specified segment
setImage:forSegment:	Sets the image for the specified segment.
setLabel:forSegment:	Sets the label for the specified segment.
setMenu:forSegment:	Sets the menu for the specified segment.
setSegmentCount:	Sets the number of segments in the receiver.
setSelected:forSegment:	Sets the selection state of the specified segment.
setSelectedSegment:	Sets the selected segment of the receiver.
setWidth:forSegment:	Sets the width of the specified segment.
widthForSegment:	Returns the width of the specified segment.

NSShadow (New)

Complete reference information is available in the NSShadow reference.

Instance Methods

init	Returns an NSShadow object initialized with default values.
set	Sets the shadow of subsequent drawing operations to the shadow represented by the receiver.
setShadowBlurRadius:	Sets the blur radius of the receiver.
setShadowColor:	Sets the shadow color for the receiver.
setShadowOffset:	Sets the offset values for the receiver.
shadowBlurRadius	Returns the blur radius of the receiver.
shadowColor	Returns the color for the receiver.
shadowOffset	Returns the offset values for the receiver.

NSSliderCell

Complete reference information is available in the `NSSliderCell` reference.

Instance Methods

`setSliderType:`	Sets the type of slider to a bar or a dial.
`sliderType`	Returns the slider type; either a bar or a dial.

NSSpeechRecognizer (New)

Complete reference information is available in the `NSSpeechRecognizer` reference.

Instance Methods

`blocksOtherRecognizers`	Returns whether the receiver should block all other recognizers (that is, other applications attempting to understand spoken commands) when listening.
`commands`	Returns an array of strings defining the commands for which the receiver should listen.
`delegate`	Returns the receiver's delegate.
`displayedCommandsTitle`	Returns the title of the commands section or nil if there is no title.
`init`	Initializes and returns an instance of the NSSpeechRecognizer class.
`listensInForegroundOnly`	Returns whether the receiver should only enable its commands when the receiver's application is the frontmost one.
`setBlocksOtherRecognizers:`	Sets whether the receiver's commands should be the only enabled commands on the system.
`setCommands:`	Sets the list of commands for which the receiver should listen to commands.
`setDelegate:`	Sets the receiver's delegate.
`setDisplayedCommandsTitle:`	Sets whether the speech-recognition commands should be displayed indented under a section title in the Speech Commands window, and if so, sets the title string to display.
`setListensInForegroundOnly:`	Sets whether the receiver should only enable its commands when the receiver's application is the frontmost one.
`startListening`	Tells the speech recognition engine to begin listening for commands.

stopListening	Tells the speech recognition engine to suspend listening for commands.

Delegate Methods

speechRecognizer:didRecognizeCommand:	Invoked when the recognition engine has recognized the application command command.

NSSpeechSynthesizer (New)

Complete reference information is available in the NSSpeechSynthesizer reference.

Class Methods

attributesForVoice:	Provides the attribute dictionary of a voice.
availableVoices	Provides the identifiers of the voices available on the system.
defaultVoice	Provides the identifier of the default voice.
isAnyApplicationSpeaking	Indicates whether any other application is currently speaking through the sound output device.

Instance Methods

delegate	Returns the receiver’s delegate.
initWithVoice:	Initializes the receiver with a voice.
isSpeaking	Indicates whether the receiver is currently generating synthesized speech.
setDelegate:	Sets the receiver’s delegate.
setUsesFeedbackWindow:	Specifies whether the receiver uses the speech feedback window.
setVoice:	Sets the receiver’s current voice.
startSpeakingString:	Begins speaking synthesized text through the system’s default sound output device.
startSpeakingString:toURL:	Begins synthesizing text into a sound (AIFF) file.
stopSpeaking	Stops synthesis in progress.
usesFeedbackWindow	Indicates whether the receiver uses the speech feedback window.
voice	Returns the identifier of the receiver’s current voice.

Delegate Methods

`speechSynthesizer:didFinishSpeaking:`	Sent when an NSSpeechSynthesizer object finishes speaking through the sound output device.
`speechSynthesizer:willSpeakPhoneme:`	Sent just before a synthesized phoneme is spoken through the sound output device.
`speechSynthesizer:willSpeakWord:ofString:`	Sent just before a synthesized word is spoken through the sound output device.

NSSpellChecker

Complete reference information is available in the `NSSpellChecker` reference.

Instance Methods

`completionsForPartialWordRange:inString:`
`language:inSpellDocumentWithTag:`
	Provides a list of complete words that the user might be trying to type based on a partial word in a given string.

NSStatusItem

Complete reference information is available in the `NSStatusItem` reference.

Instance Methods

`alternateImage`	Returns the alternate image that is displayed when a status bar item is highlighted.
`drawStatusBarBackgroundInRect:withHighlight:`	Draws the menu background pattern for a custom status-bar item in regular or highlight pattern.
`popUpStatusItemMenu:`	Displays a menu under a custom status bar item.
`setAlternateImage:`	Sets an alternate image to be displayed when a status bar item is highlighted.

NSTableColumn

Complete reference information is available in the `NSTableColumn` reference.

Instance Methods

`setSortDescriptorPrototype:`	Sets the receiver's sort descriptor prototype.
`sortDescriptorPrototype`	Returns the receiver's sort descriptor prototype.

NSTableHeaderCell

Complete reference information is available in the `NSTableHeaderCell` reference.

Instance Methods

`drawSortIndicatorWithFrame:inView:ascending:priority:`	Draws a sorting indicator given a cellFrame contained inside controlView.
`sortIndicatorRectForBounds:`	Returns the location to display the sorting indicator given theRect.

NSTableView

Complete reference information is available in the `NSTableView` reference.

Instance Methods

`drawBackgroundInClipRect:`	Draws the background in the clip rect specified by clipRect.
`gridStyleMask`	Returns the receiver's grid style mask.
`selectColumnIndexes:byExtendingSelection:`	Sets the column selection using indexes.
`selectedColumnIndexes`	Returns an index set containing the indexes of the selected columns.
`selectedRowIndexes`	Returns an index set containing the indexes of the selected rows.
`selectRowIndexes:byExtendingSelection:`	Sets the row selection using indexes.
`setGridStyleMask:`	Sets the grid style mask to specify if no grid lines, vertical grid lines, or horizontal grid lines should be displayed.
`setSortDescriptors:`	Sets the receiver's sort descriptors to the NSSortDescriptor objects in array.

`setUsesAlternatingRowBackgroundColors:`	Sets whether the receiver uses the standard alternating row colors for its background.
`sizeToFit`	Changes the width of columns in the receiver so all columns are visible.
`sortDescriptors`	Returns the receiver's sort descriptors.
`usesAlternatingRowBackgroundColors`	Returns a Boolean value that indicates whether the receiver uses the standard alternating row colors for its background.

Delegate Methods

`tableView:sortDescriptorsDidChange:`	Invoked by aTableView to indicate that sorting may need to be done.

NSTextFieldCell

Complete reference information is available in the `NSTextFieldCell` reference.

Instance Methods

`placeholderAttributedString`	Returns the cell's attributed placeholder string.
`placeholderString`	Returns the cell's plain text placeholder string.
`setPlaceholderAttributedString:`	Sets the placeholder of the cell as an attributed string.
`setPlaceholderString:`	Sets the placeholder of the cell as a plain text string.

NSTextTab

Complete reference information is available in the `NSTextTab` reference.

Instance Methods

`alignment`	Returns the text alignment of the receiver.
`initWithTextAlignment:location:options:`	Initializes a text tab with the text alignment, location, and options.
`options`	Returns the dictionary of attributes associated with the receiver.

NSTextView

Complete reference information is available in the `NSTextView` reference.

Instance Methods

`allowsDocumentBackgroundColorChange`	Returns whether the receiver allows its background color to change.
`changeAttributes:`	Changes the attributes of the current selection.
`changeDocumentBackgroundColor:`	An action method used to set the background color.
`complete:`	Invokes completion in a text view.
`completionsForPartialWordRange: indexOfSelectedItem:`	Returns an array of potential completions, in the order to be presented, representing possible word completions available from a partial word.
`defaultParagraphStyle`	Returns the receiver's default paragraph style.
`drawViewBackgroundInRect:`	Draws the background of the text view.
`insertCompletion:forPartialWordRange: movement:isFinal:`	Inserts the selected completion into the text at the appropriate location.
`linkTextAttributes`	Returns the attributes used to draw the onscreen presentation of link text.
`outline:`	Adds the outline attribute to the selected text attributes if absent; removes the attribute if present.
`performFindPanelAction:`	Performs a find panel action specified by the sender's tag.
`rangeForUserCompletion`	Returns the partial range from the most recent beginning of a word up to the insertion point.
`setAllowsDocumentBackgroundColorChange:`	Sets whether the receiver allows its background color to change.
`setDefaultParagraphStyle:`	Sets the receiver's default paragraph style.
`setLinkTextAttributes:`	Sets the attributes used to draw the onscreen presentation of link text.
`setUsesFindPanel:`	Specifies whether the receiver allows for a find panel.
`toggleBaseWritingDirection:`	Changes the base writing direction of a paragraph between left-to-right and right-to-left.
`usesFindPanel`	Returns whether the receiver allows for a find panel.

Delegate Methods

textView:completions:forPartialWordRange: indexOfSelectedItem:	Returns the actual completions for a partial word.
textView:willDisplayToolTip:forCharacterAtIndex:	Returns the actual tooltip to display.

NSToolbar

Complete reference information is available in the NSToolbar reference.

Instance Methods

selectedItemIdentifier	Returns the identifier of the receiver's currently selected item, or nil if there is no selection.
setSelectedItemIdentifier:	Sets the receiver's selected item to the specified toolbar item.

Delegate Methods

toolbarSelectableItemIdentifiers:	Sent to discover the selectable item identifiers for a toolbar.

NSUserDefaultsController (New)

Complete reference information is available in the NSUserDefaultsController reference.

Class Methods

sharedUserDefaultsController	Returns the shared instance of NSUserDefaultsController, creating it if necessary.

Instance Methods

appliesImmediately	Returns whether any changes made to bound user default properties are saved immediately.
defaults	Returns the instance of NSUserDefaults in use by the receiver.
initialValues	Returns a dictionary containing the receiver's initial default values.

`initWithDefaults:initialValues:`	Returns an initialized NSUserDefaultsController object using the NSUserDefaults instance specified in defaults and the initial default values contained in the initialValues dictionary.
`revert:`	Causes the receiver to discard any unsaved changes to bound user default properties, restoring their previous values.
`revertToInitialValues:`	Causes the receiver to discard all edits and replace the values of all the user default properties with any corresponding values in the initialValues dictionary.
`save:`	Saves the values of the receiver's user default properties.
`setAppliesImmediately:`	Sets whether any changes made to the receiver's user default properties are saved immediately.
`setInitialValues:`	Sets the receiver's initial values to initialValues.
`values`	Returns a key value coding compliant object that is used to access the user default properties.

NSView

Complete reference information is available in the `NSView` reference.

Class Methods

`defaultFocusRingType`	Returns the default focus ring type.

Instance Methods

`canBecomeKeyView`	Returns YES if the receiver can become key view, NO otherwise.
`focusRingType`	Returns the type of focus ring drawn around the receiver.
`getRectsBeingDrawn:count:`	Returns by indirection a list of non-overlapping rectangles that define the area the receiver is being asked to draw in drawRect:.
`isHidden`	Returns whether the receiver is marked as hidden.
`isHiddenOrHasHiddenAncestor`	Returns YES if the receiver is marked as hidden or has an ancestor in the view hierarchy that is marked as hidden; returns NO otherwise.
`needsToDrawRect:`	Returns whether the specified rectangle intersects any part of the area that the receiver is being asked to draw.
`setFocusRingType:`	Sets the type of focus ring to be drawn around the receiver.
`setHidden:`	Sets whether the receiver is hidden.

wantsDefaultClipping	Returns whether the Application Kit's default clipping provided to drawRect: implementations is in effect.

NSWindow

Complete reference information is available in the NSWindow reference.

Instance Methods

allowsToolTipsWhenApplicationIsInactive	Indicates whether the window can display tooltips even when the application is in the background.
contentAspectRatio	Returns the window's content aspect ratio.
contentMaxSize	Returns the maximum size of the window's content view.
contentMinSize	Returns the minimum size of the window's content view.
contentRectForFrameRect:	Returns the window's content rectangle with a given frame rectangle.
contentResizeIncrements	Returns the window's content-view resizing increments.
frameRectForContentRect:	Returns the window's frame rectangle with a given content rectangle.
setAllowsToolTipsWhenApplicationIsInactive:	Specifies whether the window can display tooltips even when the application is in the background.
setContentAspectRatio:	Sets the aspect ratio (height in relation to width) of the window's content view, constraining the dimensions of its content rectangle to integral multiples of that ratio when the user resizes it.
setContentMaxSize:	Sets the maximum size of the window's content view in the window's base coordinate system.
setContentMinSize:	Sets the minimum size of the window's content view in the window's base coordinate system.
setContentResizeIncrements:	Restricts the user's ability to resize the window so the width and height of its content view change by multiples of width and height increments.

Delegate Methods

`window:willPositionSheet:usingRect:`	Sent to the delegate just before the animation of a sheet, giving it the opportunity to return a custom location for the attachment of a sheet to a window.

NSWorkspace

Complete reference information is available in the `NSWorkspace` reference.

Instance Methods

`absolutePathForAppBundleWithIdentifier:`	Returns the absolute file-system path of an application bundle.
`launchAppWithBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifier:`	Launches the application corresponding to the specified bundleIdentifier.
`openURLs:withAppBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifiers:`	Opens one or more files from an array of URLs.

Protocols

All of the protocols with new symbols are listed alphabetically, with their new methods described.

NSGlyphStorage (New)

Complete reference information is available in the `NSGlyphStorage` reference.

Instance Methods

`attributedString`	Returns the text storage object from which the NSGlyphGenerator object procures characters for glyph generation.
`insertGlyphs:length:forStartingGlyphAtIndex:characterIndex:`	Inserts the given glyphs into the glyph cache and maps them to the specified characters.
`layoutOptions`	Returns the current layout options.
`setIntAttribute:value:forGlyphAtIndex:`	Sets a custom attribute value for a given glyph.

NSMenuItem

Complete reference information is available in the NSMenuItem reference.

Instance Methods

attributedTitle	Returns the custom title string for a menu item.
indentationLevel	Returns the menu item indentation level for the receiver.
isAlternate	Returns whether the receiver is an alternate to the previous menu item.
setAlternate:	Marks the receiver as an alternate to the previous menu item.
setAttributedTitle:	Specifies a custom string for a menu item.
setIndentationLevel:	Sets the menu item indentation level for the receiver.
setToolTip:	Sets a help tag for a menu item.
toolTip	Returns the help tag for a menu item.

C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

NSAccessibility.h

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAccessibilityRaiseBadArgumentException	Raises an error if the parameter is the wrong type or has an illegal value

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAccessibilityBoundsForRangeParameterizedAttribute	The rectangle (NSValue containing an NSRect) enclosing the specified range of characters (NSValue containing an NSRange). If the range crosses a line boundary, the returned rectangle will fully enclose all the lines of characters.
NSAccessibilityCancelAction	Cancels the operation.
NSAccessibilityCancelButtonAttribute	Element for cancel button (id)
NSAccessibilityClearButtonAttribute	Search field clear button (id)
NSAccessibilityDecrementArrowSubrole	Decrement arrow
NSAccessibilityDecrementPageSubrole	Decrement page
NSAccessibilityDefaultButtonAttribute	Element for default button (id)
NSAccessibilityDialogSubrole	Dialog
NSAccessibilityDrawerCreatedNotification	A drawer has appeared.
NSAccessibilityFloatingWindowSubrole	Floating window
NSAccessibilityIncrementArrowSubrole	Increment arrow
NSAccessibilityIncrementPageSubrole	Increment page
NSAccessibilityLineForIndexParameterizedAttribute	The line number (NSNumber) of the specified character (NSNumber)
NSAccessibilityModalAttribute	Element is the window modal (NSNumber)
NSAccessibilityNumberOfCharactersAttribute	Number of characters (NSNumber)
NSAccessibilityRaiseAction	Simulates bringing a window forward by clicking on its title bar.
NSAccessibilityRangeForIndexParameterizedAttribute	The full range of characters (NSValue containing an NSRange), including the specified character, which compose a single glyph (NSNumber)
NSAccessibilityRangeForLineParameterizedAttribute	The range of characters (NSValue containing an NSRange) corresponding to the specified line number (NSNumber).
NSAccessibilityRangeForPositionParameterizedAttribute	The range of characters (NSValue containing an NSRange) composing the glyph at the specified point (NSValue containing NSPoint)

NSAccessibilityRTFForRangeParameterizedAttribute	The RTF data (NSData) describing the specified range of characters (NSValue containing an NSRange)
NSAccessibilitySearchButtonAttribute	Search field search button (id)
NSAccessibilitySearchFieldSubrole	Search field
NSAccessibilitySearchMenuAttribute	Search field menu (id)
NSAccessibilitySheetCreatedNotification	A sheet has appeared.
NSAccessibilityStandardWindowSubrole	Standard window
NSAccessibilityStringForRangeParameterizedAttribute	The substring (NSString) specified by the range (NSValue containing an NSRange)
NSAccessibilityStyleRangeForIndexParameterizedAttribute	The full range of characters (NSValue containing an NSRange), including the specified character (NSNumber), which have the same style
NSAccessibilitySystemDialogSubrole	System dialog
NSAccessibilitySystemFloatingWindowSubrole	System floating window
NSAccessibilityVisibleCharacterRangeAttribute	Range of visible text (NSValue). Returns ranges for entire lines. For example, characters that are horizontally clipped will be reported in the visible range.

NSAlert.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAlertFirstButtonReturn	The user clicked the first (rightmost) button on the dialog or sheet.
NSAlertSecondButtonReturn	The user clicked the second button from the right edge of the dialog or sheet.
NSAlertStyle	The NSAlert class defines these alert styles.
NSAlertThirdButtonReturn	The user clicked the third button from the right edge of the dialog or sheet.

NSCriticalAlertStyle	Reserved this style for critical alerts, such as when there might be severe consequences as a result of a certain user response (for example, a “clean install” will erase all data on a volume).
NSInformationalAlertStyle	An alert used to inform the user about a current or impending event.
NSWarningAlertStyle	An alert used to warn the user about a current or impending event. The purpose is more than informational but not critical. This is the default alert style.

NSApplication.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAppKitVersionNumber10_2	The Application Kit framework included in Mac OS X v10.2.
NSAppKitVersionNumber10_2_3	The Application Kit framework included in Mac OS X v10.2.3.
NSApplicationDelegateReply	These constants indicate whether or not a copy or print operation was successful, was cancelled, or failed. These constants are used by the replyToOpenOrPrint: method.
NSApplicationDelegateReplyCancel	Indicates the user cancelled the operation.
NSApplicationDelegateReplyFailure	Indicates an error occurred processing the operation.
NSApplicationDelegateReplySuccess	Indicates the operation succeeded.

NSAttributedString.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSCursorAttributeName	NSCursor
NSDocFormatTextDocumentType	Microsoft Word document.
NSExpansionAttributeName	NSNumber containing floating point value, as log of expansion factor to be applied to glyphs
NSObliquenessAttributeName	NSNumber containing floating point value, as skew to be applied to glyphs

NSShadowAttributeName	NSShadow
NSStrikethroughColorAttributeName	NSColor
NSStrikethroughStyleAttributeName	NSNumber containing integer
NSStrokeColorAttributeName	NSColor
NSStrokeWidthAttributeName	NSNumber containing floating point value, as percent of font point size
NSToolTipAttributeName	NSString
NSUnderlineColorAttributeName	NSColor
NSUnderlinePatternDash	Draw an underline using a pattern of dashes.
NSUnderlinePatternDashDot	Draw an underline using a pattern of alternating dashes and dots.
NSUnderlinePatternDashDotDot	Draw an underline using a pattern of a dash followed by two dots.
NSUnderlinePatternDot	Draw an underline using a pattern of dots.
NSUnderlinePatternSolid	Draw a solid underline.
NSUnderlineStyleDouble	Draw an underline consisting of a double line.
NSUnderlineStyleNone	Do not draw an underline.
NSUnderlineStyleSingle	Draw an underline consisting of a single line.
NSUnderlineStyleThick	Draw an underline consisting of a thick line.
NSWordMLTextDocumentType	Microsoft Word XML (WordML schema) document.

NSBitmapImageRep.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSImageLoopCount	Identifies an NSNumber object containing the number of loops to make when animating a GIF image.

NSBrowser.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSAppKitVersionNumberWithColumnResizingBrowser`	
`NSAppKitVersionNumberWithContinuousScrollingBrowser`	
`NSBrowserAutoColumnResizing`	All columns have the same width, calculated using a combination of the minimum column width and maximum number of visible columns settings.
`NSBrowserAuxiliaryOpaque`	A private data structure used internally by NSBrowser.
`NSBrowserColumnConfigurationDidChangeNotification`	Notifies the delegate when the width of a browser column has changed.
`NSBrowserColumnResizingType`	Describe types of browser column resizing.
`NSBrowserNoColumnResizing`	Neither NSBrowser nor the user can change the column width.
`NSBrowserUserColumnResizing`	The developer chooses the initial column widths, but users can resize all columns simultaneously or each column individually.

NSButtonCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSDisclosureBezelStyle`	A bezel style for use with a disclosure triangle.
`NSHelpButtonBezelStyle`	A round button with a question mark providing the standard help button look.
`NSTexturedSquareBezelStyle`	A bezel style appropriate for use with textured (metal) windows.

NSCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSBlueControlTint	Aqua control tint
NSGraphiteControlTint	Graphite control tint
NSMiniControlSize	The control has a smaller size than NSSmallControlSize.

NSColorList.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSColorListAuxiliaryOpaque	A private data structure used internally by NSColorList.

NSCursor.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAppKitVersionNumberWithCursorSizeSupport	The specific version of the AppKit framework that introduced support for cursors larger than 16 x 16 pixels in size.

NSFontDescriptor.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSFontColorAttribute	An optional NSData object that specifies the font color.

NSFontFaceAttribute	An optional NSString object that specifies the font face.
NSFontFamilyAttribute	An optional NSString object that specifies the font family.
NSFontNameAttribute	An optional NSString object that specifies the font name.
NSFontSizeAttribute	An optional NSString object, containing a float value, that specifies the font size.
NSFontVisibleNameAttribute	An optional NSString object that specifies the font's visible name.

NSFontManager.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSFontCollectionApplicationOnlyMask	Makes the collection available only to the application. This option is not yet implemented.

NSFontPanel.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSFontPanelAllModesMask	Display all the available adornments.
NSFontPanelCollectionModeMask	Display the font collections column.
NSFontPanelFaceModeMask	Display the typeface column.
NSFontPanelSizeModeMask	Display the font size column.
NSFontPanelStandardModesMask	Display the standard default font panel—that is, including the collections, typeface, and size columns.

NSGlyphGenerator.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSShowControlGlyphs`	Generates displayable glyphs for control characters.
`NSShowInvisibleGlyphs`	Generates displayable glyphs for invisible characters.
`NSWantsBidiLevels`	Generates directional formatting codes for bidirectional text.

NSGraphics.h

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSDisableScreenUpdates`	Disables screen updates.
`NSEnableScreenUpdates`	Enables screen updates
`NSShowAnimationEffect`	Runs a system animation effect.

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSAnimationEffect`	This type defines the standard system animation effects, which include both display and sound.
`NSAnimationEffectDisappearingItemDefault`	The default effect.
`NSAnimationEffectPoof`	An effect showing a puff of smoke.
`NSFocusRingType`	The focus ring type is used by NSView and NSCell to configure if and how a control should draw its focus ring.
`NSFocusRingTypeDefault`	The default focus ring type for NSView or NSCell.
`NSFocusRingTypeExterior`	The standard Aqua focus ring.
`NSFocusRingTypeNone`	No focus ring. If you set the focus ring type to this value, NSView and NSCell will not draw any focus ring.

NSKeyValueBinding.h

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSIsControllerMarker	Tests whether a given object is special marker object used for indicating the state of a selection in relation to a key.

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSMultipleValuesMarker	This marker indicates that a key's value contains multiple values that differ.
NSNoSelectionMarker	This marker indicates that the controller's selection is currently empty.
NSNotApplicableMarker	This marker indicates that an object is not key-value coding compliant for the requested key.

NSLayoutManager.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSTypesetterBehavior_10_3	The typesetter behavior introduced in Mac OS X version 10.3.

NSMenu.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSMenuDidEndTrackingNotification	Posted when menu tracking ends, even if no action is sent.

NSNib.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSNibOwner	The external object that is responsible for the instantiated nib.
NSNibTopLevelObjects	An NSMutableArray object that, if present, is populated with the top-level objects of the newly instantiated nib.

NSOpenGL.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSOpenGLPFAPixelBuffer	A Boolean attribute. If present, this attribute indicates that rendering to a pixel buffer is enabled.

NSParagraphStyle.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

alignment	
NSTabColumnTerminatorsAttributeName	The value is an NSCharacterSet object.
unused	

NSSavePanel.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSSavePanelAuxiliaryOpaque	A private data structure used internally by NSSavePanel.

NSSearchFieldCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSSearchFieldClearRecentsMenuItemTag	Identifies the menu item for clearing the current set of recent string searches in the menu.
NSSearchFieldNoRecentsMenuItemTag	Identifies the menu item that describes a lack of recent search strings (for example, “No recent searches”).
NSSearchFieldRecentsMenuItemTag	Identifies where recent search strings should appear in the “recents” menu group.
NSSearchFieldRecentsTitleMenuItemTag	Identifies the menu item that is the title of the menu group for recent search strings.

NSSegmentedCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSSegmentSwitchTracking	An NSSegmentedCell object uses the following constants, which describe the various tracking modes for a cell. You access these values using setTrackingMode: and trackingMode methods of the class.
NSSegmentSwitchTrackingMomentary	A segment is selected only when tracking.
NSSegmentSwitchTrackingSelectAny	Any segment can be selected.
NSSegmentSwitchTrackingSelectOne	Only one segment may be selected.

NSSliderCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSCircularSlider	A circular slider; that is, a dial.
NSLinearSlider	A bar-shaped slider.
NSSliderType	Define the types of sliders, used by setSliderType: and sliderType.

NSSpeechSynthesizer.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSVoiceAge	The perceived age (in years) of the voice.
NSVoiceDemoText	A demonstration string to speak.
NSVoiceGender	The perceived gender of the voice. May be either NSVoiceGenderNeuter, NSVoiceGenderFemale, or NSVoiceGenderMale.
NSVoiceGenderFemale	A female voice
NSVoiceGenderMale	A male voice
NSVoiceGenderNeuter	A neutral voice (neither male nor female)
NSVoiceIdentifier	A unique string identifying the voice. The identifiers of the system voices are listed in Listing 1.
NSVoiceLanguage	The language of the voice (currently US English only).
NSVoiceName	The name of the voice suitable for display.

NSTableView.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSTableViewGridNone	Specifies that no grid lines should be displayed.
NSTableViewSolidHorizontalGridLineMask	Specifies that horizontal grid lines should be displayed.
NSTableViewSolidVerticalGridLineMask	Specifies that vertical grid lines should be displayed.

NSText.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSCancelTextMovement`	The user cancelled the completion.
`NSOtherTextMovement`	The user performed some undefined action.

NSTextView.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSFindPanelAction`	These constants define the tags for performFindPanelAction:.
`NSFindPanelActionNext`	Finds the next instance of the queried text.
`NSFindPanelActionPrevious`	Finds the previous instance of the queried text.
`NSFindPanelActionReplace`	Replaces a single query instance within the text view.
`NSFindPanelActionReplaceAll`	Replaces all query instances within the text view.
`NSFindPanelActionReplaceAllInSelection`	Replaces all query instances within the selection.
`NSFindPanelActionReplaceAndFind`	Replaces a single query instance and finds the next.
`NSFindPanelActionSetFindString`	Sets the query string to the current selection.
`NSFindPanelActionShowFindPanel`	Displays the find panel.
`NSTextViewDidChangeTypingAttributesNotification`	Posted when there is a change in the typing attributes within a text view.

NSWindow.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAppKitVersionNumberWithCustomSheetPosition	

NSWorkspace.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSWorkspaceDidWakeNotification	Posted when the machine wakes from sleep.
NSWorkspaceLaunchAllowingClassicStartup	Start up the Classic compatibility environment, if it is required by the application.
NSWorkspaceLaunchAndHide	Tell the application to hide itself as soon as it has finished launching.
NSWorkspaceLaunchAndHideOthers	Hide all applications except the newly launched one.
NSWorkspaceLaunchAndPrint	Print items instead of opening them.
NSWorkspaceLaunchAsync	Launch the application and return the results asynchronously.
NSWorkspaceLaunchDefault	Launch the application asynchronously and launch it in the Classic environment, if required.
NSWorkspaceLaunchInhibitingBackgroundOnly	Causes launch to fail if the target is background-only.
NSWorkspaceLaunchNewInstance	Create a new instance of the application, even if one is already running.
NSWorkspaceLaunchOptions	Constants representing different launch options.
NSWorkspaceLaunchPreferringClassic	Force the application to launch in the Classic compatibility environment.

NSWorkspaceLaunchWithoutActivation	Launch the application but do not bring it into the foreground.
NSWorkspaceLaunchWithoutAddingToRecents	Do not add the application or documents to the Recents menu.
NSWorkspaceSessionDidBecomeActiveNotification	Posted after a user session is switched in.
NSWorkspaceSessionDidResignActiveNotification	Posted before a user session is switched out.
NSWorkspaceWillSleepNotification	Posted before the machine goes to sleep.

10.2 Symbol Changes

This article lists the symbols added to `AppKit.framework` in Mac OS X v10.2.

Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

NSAppleScript

Complete reference information is available in the `NSAppleScript` reference.

Instance Methods

`richTextSource`	Returns the syntax-highlighted source code of the receiver if the receiver has been compiled and its source code is available.

NSBitmapImageRep

Complete reference information is available in the `NSBitmapImageRep` reference.

Instance Methods

`incrementalLoadFromData:complete:`	Loads the current image data into an incrementally-loaded image representation and returns the current status of the image.
`initForIncrementalLoad`	Initializes and returns the receiver, a newly allocated NSBitmapImageRep object, for incremental loading.

NSBrowserCell

Complete reference information is available in the `NSBrowserCell` reference.

Instance Methods

`image`	Returns the receiver's image.
`setImage:`	Sets the receiver's image, retaining the image.

NSColor

Complete reference information is available in the `NSColor` reference.

Class Methods

`alternateSelectedControlColor`	Returns the system color used for the face of a selected control.
`alternateSelectedControlTextColor`	Returns the system color used for text in a selected control.

NSFont

Complete reference information is available in the `NSFont` reference.

Instance Methods

`coveredCharacterSet`	Returns an NSCharacterSet object containing all of the nominal characters renderable by the receiver, which is all of the entries mapped in the receiver's 'cmap' table.

NSGlyphInfo (New)

Complete reference information is available in the `NSGlyphInfo` reference.

Class Methods

`glyphInfoWithCharacterIdentifier:collection:baseString:`	Instantiates and returns an NSGlyphInfo object using a character identifier and a character collection.
`glyphInfoWithGlyph:forFont:baseString:`	Instantiates and returns an NSGlyphInfo object using a glyph index and a specified font.
`glyphInfoWithGlyphName:forFont:baseString:`	Instantiates and returns an NSGlyphInfo object using a glyph name and a specified font.

Instance Methods

characterCollection	Returns an NSCharacterCollection value specifying the glyph–to–character identifier mapping of the receiver.
characterIdentifier	Returns the receiver's character identifier (CID).
glyphName	Returns the receiver's glyph name.

NSGraphicsContext

Complete reference information is available in the NSGraphicsContext reference.

Instance Methods

patternPhase	Returns the amount to offset the pattern color when filling the receiver.
setPatternPhase:	Sets the amount to offset the pattern color when filling the receiver.

NSImage

Complete reference information is available in the NSImage reference.

Instance Methods

cacheMode	Returns the receiver's caching mode.
cancelIncrementalLoad	Cancels the current download operation immediately, if the image is being incrementally loaded.
initByReferencingURL:	Initializes and returns an NSImage instance and associates it with the specified URL.
setCacheMode:	Set the receiver's caching mode.

Delegate Methods

image:didLoadPartOfRepresentation:withValidRows:	During incremental loading, this method is called repeatedly to inform the delegate that more of the image data is available.

`image:didLoadRepresentation:withStatus:`	For incremental loading, this method is invoked when the specified image has been loaded and decompressed as fully as is possible.
`image:didLoadRepresentationHeader:`	During incremental loading, this method is called once enough data has been read to determine the size of the image.
`image:willLoadRepresentation:`	For incremental loading, this method is invoked when you first attempt to draw the image or otherwise access the bitmap data.

NSLayoutManager

Complete reference information is available in the `NSLayoutManager` reference.

Instance Methods

`defaultLineHeightForFont:`	Returns the default line height for a line of text drawn using a given font.
`getGlyphsInRange:glyphs:characterIndexes:glyphInscriptions:elasticBits:bidiLevels:`	Returns the glyphs and information needed to perform layout for the given glyph range.
`setTypesetterBehavior:`	Sets the default typesetter behavior.
`typesetterBehavior`	Returns the current typesetter behavior.

NSMenu

Complete reference information is available in the `NSMenu` reference.

Class Methods

`menuBarVisible`	Returns a Boolean value that indicates whether the menu bar is visible.
`setMenuBarVisible:`	Sets whether the menu bar is visible and selectable by the user.

NSMutableParagraphStyle

Complete reference information is available in the `NSMutableParagraphStyle` reference.

Instance Methods

`setBaseWritingDirection:`	Sets the base writing direction for the receiver.

NSObject

Complete reference information is available in the `NSObject` reference.

Instance Methods

`accessibilityActionDescription:`	Returns a localized description of the action action.
`accessibilityActionNames`	Returns an array of action names supported by the receiver.
`accessibilityAttributeNames`	Returns an array of attribute names supported by the receiver.
`accessibilityAttributeValue:`	Returns the value of the receiver's attribute attribute.
`accessibilityFocusedUIElement`	Returns the deepest descendant of the user interface hierarchy that has the focus.
`accessibilityHitTest:`	Returns the deepest descendant of the accessibility hierarchy that contains the point point, which is in lower-left relative screen coordinates.
`accessibilityIsAttributeSettable:`	Returns YES if the value for the receiver's attribute attribute can be set.
`accessibilityIsIgnored`	Returns YES if the receiver should not be visible in the parent-child accessibility hierarchy.
`accessibilityPerformAction:`	Performs the action associated with the name action.
`accessibilitySetValue:forAttribute:`	Sets the value of the receiver's attribute attribute to value.
`namesOfPromisedFilesDroppedAtDestination:`	Returns the names of the files that the receiver promises to create at a specified location.

NSOpenGLContext

Complete reference information is available in the `NSOpenGLContext` reference.

Instance Methods

createTexture:fromView:internalFormat:	Creates a new texture from the contents of the specified view.
currentVirtualScreen	Returns the current virtual screen for the receiver.
setCurrentVirtualScreen:	Sets the current virtual screen for the receiver.

NSParagraphStyle

Complete reference information is available in the NSParagraphStyle reference.

Class Methods

defaultWritingDirectionForLanguage:	Returns the default writing direction for the specified language.

Instance Methods

baseWritingDirection	Returns the base writing direction for the receiver.

NSPrintInfo

Complete reference information is available in the NSPrintInfo reference.

Instance Methods

imageablePageBounds	Returns the imageable area of a sheet of paper specified by the receiver.

NSPrintOperation

Complete reference information is available in the NSPrintOperation reference.

Instance Methods

jobStyleHint	Returns the type of content that the print job is printing.
setJobStyleHint:	Sets the type of content that the print job is printing.

NSPrintPanel

Complete reference information is available in the `NSPrintPanel` reference.

Instance Methods

`jobStyleHint`	Returns the type of content that the Print panel is representing.
`setJobStyleHint:`	Sets the type of content the Print panel is representing.

NSProgressIndicator

Complete reference information is available in the `NSProgressIndicator` reference.

Instance Methods

`isDisplayedWhenStopped`	Returns a Boolean value indicating whether the receiver shows itself even when it's not animating.
`setDisplayedWhenStopped:`	Sets whether the receiver hides itself when it isn't animating.
`setStyle:`	Sets the style of the progress indicator (bar or spinning).
`sizeToFit`	This action method resizes the receiver to an appropriate size depending on what style returns.
`style`	Returns the style of the progress indicator (bar or spinning).

NSResponder

Complete reference information is available in the `NSResponder` reference.

Instance Methods

`shouldBeTreatedAsInkEvent:`	Returns YES if the specified event should be treated as an ink event, NO if it should be treated as a mouse event.

NSTextField

Complete reference information is available in the `NSTextField` reference.

Instance Methods

`bezelStyle`	Returns the receiver's bezel style.
`setBezelStyle:`	Sets the receiver's bezel style.

NSTextFieldCell

Complete reference information is available in the `NSTextFieldCell` reference.

Instance Methods

`bezelStyle`	Returns the receiver's bezel style.
`setBezelStyle:`	Sets the receiver's bezel style.

NSTextView

Complete reference information is available in the `NSTextView` reference.

Instance Methods

`acceptsGlyphInfo`	Returns whether the receiver accepts the glyph info attribute.
`setAcceptsGlyphInfo:`	Sets whether the receiver accepts the glyph info attribute.

NSToolbar

Complete reference information is available in the `NSToolbar` reference.

Instance Methods

`setSizeMode:`	Sets the receiver's size mode.
`sizeMode`	Returns the receiver's size mode.

NSTypesetter

Complete reference information is available in the `NSTypesetter` reference.

Class Methods

defaultTypesetterBehavior	Returns the default typesetter behavior.
sharedSystemTypesetterForBehavior:	Returns a shared instance of a reentrant typesetter that implements typesetting with the specified behavior.

NSView

Complete reference information is available in the NSView reference.

Instance Methods

dragPromisedFilesOfTypes:fromRect: source:slideBack:event:	Initiates a dragging operation from the receiver, allowing the user to drag one or more promised files (or directories) into any application that has window or view objects that accept promised file data.
mouseDownCanMoveWindow	Returns YES if the receiver does not need to handle a mouse down and can pass it through to superviews; NO if it needs to handle the mouse down.

NSWindow

Complete reference information is available in the NSWindow reference.

Class Methods

standardWindowButton:forStyleMask:	Returns a new instance of a given standard window button, sized appropriately for a given window style.

Instance Methods

addChildWindow:ordered:	Adds a given window as a child window of the window.
childWindows	Returns an array of the window's attached child windows.
ignoresMouseEvents	Indicates whether the window is transparent to mouse events.
invalidateShadow	Invalidates the window shadow so that it is recomputed based on the current window shape.
isMovableByWindowBackground	Indicates whether the window is movable by clicking and dragging anywhere in its background.

parentWindow	Returns the parent window to which the window is attached as a child.
removeChildWindow:	Detaches a given child window from the window.
setIgnoresMouseEvents:	Specifies whether the window is transparent to mouse clicks and other mouse events, allowing overlay windows.
setMovableByWindowBackground:	Sets whether the window is movable by clicking and dragging anywhere in its background.
setParentWindow:	Adds the window as a child of a given window. For use by subclasses when setting the parent window in the window.
standardWindowButton:	Returns the window button of a given window button kind in the window's view hierarchy.

NSWorkspace

Complete reference information is available in the NSWorkspace reference.

Instance Methods

activeApplication	Returns a dictionary with information about the current active application.
launchedApplications	Returns an array of dictionaries, one entry for each running application.

Protocols

All of the protocols with new symbols are listed alphabetically, with their new methods described.

NSDraggingInfo

Complete reference information is available in the NSDraggingInfo reference.

Instance Methods

namesOfPromisedFilesDroppedAtDestination:	Sets the drop location for promised files and returns the names of the files that the receiver promises to create there.

C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

NSAccessibility.h

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAccessibilityPostNotification	Sends a notification to any observing assistive applications.
NSAccessibilityUnignoredAncestor	Returns an unignored accessibility object, ascending the hierarchy if necessary.
NSAccessibilityUnignoredChildren	Returns a list of unignored accessibility objects, descending the hierarchy if necessary.
NSAccessibilityUnignoredChildrenForOnlyChild	Returns a list of unignored accessibility objects, descending the hierarchy if necessary.
NSAccessibilityUnignoredDescendant	Returns an unignored accessibility object, descending the hierarchy if necessary.

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAccessibilityApplicationActivatedNotification	Application has activated.
NSAccessibilityApplicationDeactivatedNotification	Application has deactivated.
NSAccessibilityApplicationHiddenNotification	Application has been hidden.
NSAccessibilityApplicationRole	Application
NSAccessibilityApplicationShownNotification	Application has been shown.
NSAccessibilityBrowserRole	Browser
NSAccessibilityBusyIndicatorRole	Busy indicator
NSAccessibilityButtonRole	Button
NSAccessibilityCheckBoxRole	Checkbox

NSAccessibilityChildrenAttribute	Elements that current element contains (NSArray)
NSAccessibilityCloseButtonAttribute	Element for close button (id)
NSAccessibilityCloseButtonSubrole	Window's close button
NSAccessibilityColorWellRole	Color well
NSAccessibilityColumnRole	Column
NSAccessibilityColumnsAttribute	Table's columns (NSArray)
NSAccessibilityColumnTitlesAttribute	Elements for column titles (NSArray)
NSAccessibilityComboBoxRole	Combo box
NSAccessibilityConfirmAction	Simulates pressing Return in object, such as a text field.
NSAccessibilityContentsAttribute	Elements that represent the contents in the current element, such as the document view of a scroll view (NSArray)
NSAccessibilityDecrementAction	Decrements the value of object.
NSAccessibilityDecrementButtonAttribute	Stepper's decrement button (id)
NSAccessibilityDisclosedByRowAttribute	Row disclosing this row (id)
NSAccessibilityDisclosedRowsAttribute	Rows disclosed by this row (NSArray)
NSAccessibilityDisclosingAttribute	Flag indicating whether a row is disclosing other rows (NSNumber)
NSAccessibilityDocumentAttribute	URL for the file represented by the element (NSString)
NSAccessibilityDrawerRole	Drawer
NSAccessibilityEditedAttribute	Flag indicating whether element has been modified (NSNumber)
NSAccessibilityEnabledAttribute	Flag indicating enabled state of element (NSNumber)
NSAccessibilityErrorCodeExceptionInfo	Integer error code used for debugging (as an NSNumber).
NSAccessibilityExpandedAttribute	Flag indicating whether element is expanded (NSNumber)
NSAccessibilityFilenameAttribute	Filename associated with element (NSString)

NSAccessibilityFocusedAttribute	Flag indicating presence of keyboard focus (NSNumber)
NSAccessibilityFocusedUIElementAttribute	Element with the current focus (id)
NSAccessibilityFocusedUIElementChangedNotification	Element has gained focus.
NSAccessibilityFocusedWindowAttribute	Application's window with current focus (id)
NSAccessibilityFocusedWindowChangedNotification	Key window has changed.
NSAccessibilityFrontmostAttribute	Flag indicating whether application is in front (NSNumber)
NSAccessibilityGroupRole	Group
NSAccessibilityGrowAreaAttribute	Element for grow area (id)
NSAccessibilityGrowAreaRole	Window's grow (resize) area
NSAccessibilityHeaderAttribute	Table view's header element (id)
NSAccessibilityHelpAttribute	Instance description, such as a help tag string (NSString)
NSAccessibilityHiddenAttribute	Flag indicating whether application is hidden (NSNumber)
NSAccessibilityHorizontalOrientationValue	Element is oriented horizontally
NSAccessibilityHorizontalScrollBarAttribute	Scroll view's horizontal scroll bar (id)
NSAccessibilityImageRole	Image
NSAccessibilityIncrementAction	Increments the value of object.
NSAccessibilityIncrementButtonAttribute	Stepper's increment button (id)
NSAccessibilityIncrementorRole	Stepper
NSAccessibilityListRole	List
NSAccessibilityMainAttribute	Flag indicating whether a window is the main window (NSNumber)
NSAccessibilityMainWindowAttribute	Application's main window (id)
NSAccessibilityMainWindowChangedNotification	Main window has changed.
NSAccessibilityMaxValueAttribute	Element's maximum value (id)
NSAccessibilityMenuBarAttribute	Application's menu bar (id)
NSAccessibilityMenuBarRole	Menu bar

`NSAccessibilityMenuButtonRole`	Menu button
`NSAccessibilityMenuItemRole`	Menu item
`NSAccessibilityMenuRole`	Menu
`NSAccessibilityMinimizeButtonAttribute`	Element for minimize button (id)
`NSAccessibilityMinimizeButtonSubrole`	Window's minimize button
`NSAccessibilityMinimizedAttribute`	Flag indicating whether a window is minimized (NSNumber)
`NSAccessibilityMinValueAttribute`	Element's minimum value (id)
`NSAccessibilityNextContentsAttribute`	Contents following current divider element, such as a subview adjacent to a split view's splitter element (NSArray)
`NSAccessibilityOrientationAttribute`	Element's orientation, which can have the value NSAccessibilityHorizontal-OrientationValue or NSAccessibilityVertical-OrientationValue
`NSAccessibilityOutlineRole`	Outline
`NSAccessibilityOutlineRowSubrole`	Outline row
`NSAccessibilityOverflowButtonAttribute`	Toolbar's overflow button (id)
`NSAccessibilityParentAttribute`	Element containing current element (id)
`NSAccessibilityPickAction`	Selects the object, such as a menu item.
`NSAccessibilityPopUpButtonRole`	Pop-up button
`NSAccessibilityPositionAttribute`	Screen position of element's lower-left corner in lower-left relative screen coordinates (NSValue)
`NSAccessibilityPressAction`	Simulates clicking an object, such as a button.
`NSAccessibilityPreviousContentsAttribute`	Contents preceding current divider element, such as a subview adjacent to a split view's splitter bar element (NSArray)
`NSAccessibilityProgressIndicatorRole`	Progress indicator
`NSAccessibilityProxyAttribute`	Element for title's icon (id)

`NSAccessibilityRadioButtonRole`	Radio button
`NSAccessibilityRadioGroupRole`	Radio group
`NSAccessibilityRelevanceIndicatorRole`	
`NSAccessibilityRoleAttribute`	Element type, such as NSAccessibilityRadioButtonRole (NSString). See the role attribute table below
`NSAccessibilityRoleDescriptionAttribute`	Localized, user-readable description of role, such as radio button (NSString)
`NSAccessibilityRowRole`	Row
`NSAccessibilityRowsAttribute`	Table's rows (NSArray)
`NSAccessibilityScrollAreaRole`	Scroll view
`NSAccessibilityScrollBarRole`	Scroll bar
`NSAccessibilitySecureTextFieldSubrole`	Secure text field
`NSAccessibilitySelectedAttribute`	Flag indicating whether element is selected (NSNumber)
`NSAccessibilitySelectedChildrenAttribute`	Currently selected children (NSArray)
`NSAccessibilitySelectedColumnsAttribute`	Table's selected columns (NSArray)
`NSAccessibilitySelectedRowsAttribute`	Table's selected rows (NSArray)
`NSAccessibilitySelectedTextAttribute`	Currently selected text (NSString)
`NSAccessibilitySelectedTextRangeAttribute`	Range of selected text (NSValue)
`NSAccessibilitySheetRole`	Sheet
`NSAccessibilitySizeAttribute`	Element size (NSValue)
`NSAccessibilitySliderRole`	Slider
`NSAccessibilitySortDirectionAttribute`	Column's sort direction (NSString). See "Column sort direction".
`NSAccessibilitySplitGroupRole`	Split view
`NSAccessibilitySplitterRole`	Splitter bar of a split view
`NSAccessibilitySplittersAttribute`	Splitter bar elements in a split view (NSArray)
`NSAccessibilityStaticTextRole`	Uneditable text

NSAccessibilitySubroleAttribute	Element subrole type, such as NSAccessibilityTableRowSubrole (NSString). See the subrole attribute table below.
NSAccessibilitySystemWideRole	Special system-wide element
NSAccessibilityTabGroupRole	Tab group
NSAccessibilityTableRole	Table
NSAccessibilityTableRowSubrole	Table row
NSAccessibilityTabsAttribute	Tab view's tab elements (NSArray)
NSAccessibilityTextAreaRole	Text view
NSAccessibilityTextFieldRole	Text field
NSAccessibilityTitleAttribute	Title of element, such as button text (NSString)
NSAccessibilityTitleUIElementAttribute	Element's title element (id)
NSAccessibilityToolbarButtonAttribute	Element for toolbar toggle button (id)
NSAccessibilityToolbarButtonSubrole	Window's toolbar toggle button
NSAccessibilityToolbarRole	Toolbar
NSAccessibilityUIElementDestroyedNotification	Element has been destroyed.
NSAccessibilityUnknownRole	Unknown object type
NSAccessibilityUnknownSubrole	Unknown subrole
NSAccessibilityValueAttribute	Element's value (id)
NSAccessibilityValueChangedNotification	Element's value has changed.
NSAccessibilityValueIndicatorRole	Value indicator
NSAccessibilityVerticalOrientationValue	Element is oriented vertically
NSAccessibilityVerticalScrollBarAttribute	Scroll view's vertical scroll bar (id)
NSAccessibilityVisibleChildrenAttribute	Child elements that are visible (NSArray)
NSAccessibilityVisibleColumnsAttribute	Table's visible columns (NSArray)
NSAccessibilityVisibleRowsAttribute	Table's visible rows (NSArray)
NSAccessibilityWindowAttribute	Window containing current element (id)

NSAccessibilityWindowCreatedNotification	A new window has appeared.
NSAccessibilityWindowDeminiaturizedNotification	Window has been restored to full size from the Dock.
NSAccessibilityWindowMiniaturizedNotification	Window has been put in the Dock.
NSAccessibilityWindowMovedNotification	Window has moved.
NSAccessibilityWindowProxyRole	
NSAccessibilityWindowResizedNotification	Window has changed size.
NSAccessibilityWindowRole	Window
NSAccessibilityWindowsAttribute	Application's windows (NSArray)
NSAccessibilityWindowTitleRole	
NSAccessibilityZoomButtonAttribute	Element for zoom button (id)
NSAccessibilityZoomButtonSubrole	Window's zoom button

NSApplication.h

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSApplicationLoad	Startup function to call when running Cocoa code from a Carbon application.

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAppKitVersionNumber10_1	The Application Kit framework included in Mac OS X v10.1.

NSAttributedString.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSGlyphInfoAttributeName	The name of an NSGlyphInfo object.

NSBitmapImageRep.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSImageCurrentFrame	Identifies an NSNumber object containing the current frame for an animated GIF file.
NSImageCurrentFrameDuration	Identifies an NSNumber object containing the duration (in seconds) of the current frame for an animated GIF image.
NSImageFrameCount	Identifies an NSNumber object containing the number of frames in an animated GIF file.
NSImageRepLoadStatus	These constants represent the various status values returned by incrementalLoadFromData:complete:.
NSImageRepLoadStatusCompleted	Enough data has been provided to successfully decompress the image (regardless of the complete: flag).
NSImageRepLoadStatusInvalidData	An error occurred during image decompression. The image contains the portions of the data that have already been successfully decompressed, if any
NSImageRepLoadStatusReadingHeader	The image format is known, but not enough data has been read to determine the size, depth, etc., of the image. You should continue to provide more data.
NSImageRepLoadStatusUnexpectedEOF	incrementalLoadFromData:complete: was called with YES, but not enough data was available for decompression. The image contains the portions of the data that have already been successfully decompressed, if any.
NSImageRepLoadStatusUnknownType	Not enough data to determine image format. You should continue to provide more data.
NSImageRepLoadStatusWillNeedAllData	Incremental loading cannot be supported.

NSColor.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAppKitVersionNumberWithPatternColorLeakFix	The specific version of the AppKit framework that introduced the fix for correctly autoreleasing objects returned by the colorWithPatternImage: method.

NSColorPanel.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSColorPanelCrayonModeMask	Crayons.
NSCrayonModeColorPanel	Crayons.

NSErrors.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAccessibilityException	

NSGlyphInfo.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAdobeCNS1CharacterCollection	Indicates the Adobe-CNS1 mapping.
NSAdobeGB1CharacterCollection	Indicates the Adobe-GB1 mapping.
NSAdobeJapan1CharacterCollection	Indicates the Adobe-Japan1 mapping.
NSAdobeJapan2CharacterCollection	Indicates the Adobe-Japan2 mapping.
NSAdobeKorea1CharacterCollection	Indicates the Adobe-Korea1 mapping.

NSCharacterCollection	The following values specify the mapping of character identifiers to glyphs, and are returned by characterCollection.
NSIdentityMappingCharacterCollection	Indicates that the character identifier is equal to the glyph index.

NSImage.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSImageCacheAlways	Always generate a cache when drawing.
NSImageCacheBySize	Cache if cache size is smaller than the original data.
NSImageCacheDefault	Caching is unspecified.
NSImageCacheMode	These constants specify the caching policy on a per NSImage basis. The caching policy is set using cacheMode and setCacheMode:.
NSImageCacheNever	Never cache; always draw direct.
NSImageLoadStatus	These constants are status values passed to the incremental loading delegate method image:didLoadRepresentation: withStatus:.
NSImageLoadStatusCancelled	Image loading was canceled.
NSImageLoadStatusCompleted	Enough data has been provided to completely decompress the image.
NSImageLoadStatusInvalidData	An error occurred during image decompression.
NSImageLoadStatusReadError	Not enough data was available for full decompression of the image.
NSImageLoadStatusUnexpectedEOF	Not enough data was available for full decompression of the image.

NSLayoutManager.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSGlyphAttributeBidiLevel`	The bidirectional level (direction) of the glyph.
`NSTypesetterBehavior`	These constants define the behavior of NSLayoutManager and NSTypesetter when laying out lines. They are used by setTypesetterBehavior: and typesetterBehavior to control the compatibility level of the typesetter.
`NSTypesetterBehavior_10_2`	The typesetter behavior introduced in Mac OS X version 10.2. This typesetter behavior provides enhanced line and character spacing accuracy and supports more languages than the original typesetter behavior.
`NSTypesetterBehavior_10_2_WithCompatibility`	Typesetting same as NSTypesetterBehavior_10_2 but using line widths and height metric calculations that are the same as with NSTypesetterOriginalBehavior.
`NSTypesetterLatestBehavior`	The most current typesetter behavior in the current system version.
`NSTypesetterOriginalBehavior`	The original typesetter behavior, as shipped with Mac OS X v10.1 and earlier.

NSOpenGL.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSOpenGLCPSurfaceOpacity`	Set or get the surface opacity.
`NSOpenGLCPSurfaceOrder`	Get or set the surface order.
`NSOpenGLPFAAuxDepthStencil`	Each auxiliary buffer has its own depth stencil.
`NSOpenGLPFASampleBuffers`	Value is a nonnegative number indicating the number of multisample buffers.

NSOpenGLPFASamples	Value is a nonnegative indicating the number of samples per multisample buffer.
NSOpenGLPFAVirtualScreenCount	The number of virtual screens in this format.

NSPanel.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSNonactivatingPanelMask	The panel can receive keyboard input without activating the owning application.

NSParagraphStyle.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSWritingDirection	These constants specify the writing directions:
NSWritingDirectionLeftToRight	The writing direction is left to right.
NSWritingDirectionRightToLeft	The writing direction is right to left.

NSPasteboard.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSFilesPromisePboardType	Promised files.
NSVCardPboardType	VCard data

NSPrintPanel.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSPrintPhotoJobStyleHint`	Output contains photographic data.

NSProgressIndicator.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSProgressIndicatorBarStyle`	A rectangular indicator that can be determinate or indeterminate.
`NSProgressIndicatorSpinningStyle`	A small square indicator that can be indeterminate only .
`NSProgressIndicatorStyle`	Specify the progress indicator’s style and are used by style and setStyle:.

NSTabView.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSAppKitVersionNumberWithDirectionalTabs`	

NSTabViewItem.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSTabViewItemAuxiliaryOpaque`	A private data structure used by NSTabViewItem.

NSTextFieldCell.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSTextFieldBezelStyle`	Specify the bezel style of the text field cell.
`NSTextFieldRoundedBezel`	Corners are rounded.
`NSTextFieldSquareBezel`	Corners are square.

NSToolbar.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSToolbarSizeMode`	These constants specify toolbar display modes and are used by sizeMode and setSizeMode:.
`NSToolbarSizeModeDefault`	The toolbar uses the system-defined default size, which is NSToolbarSizeModeRegular.
`NSToolbarSizeModeRegular`	The toolbar uses regular-sized controls and 32 by 32 pixel icons.
`NSToolbarSizeModeSmall`	The toolbar uses small-sized controls and 24 by 24 pixel icons.

NSTypesetter.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`curBidiLevel`	
`glyphBidiLevelCache`	

NSWindow.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSTexturedBackgroundWindowMask`	The window displays with a metal-textured background. Additionally, the window may be moved by clicking and dragging anywhere in the window background. A bordered window with this mask gets rounded bottom corners.
`NSWindowButton`	These constants provide a way to access standard title bar buttons:
`NSWindowCloseButton`	The close button.
`NSWindowDocumentIconButton`	The document icon button.
`NSWindowMiniaturizeButton`	The minimize button.
`NSWindowToolbarButton`	The toolbar button.
`NSWindowZoomButton`	The zoom button.

10.1 Symbol Changes

This article lists the symbols added to AppKit.framework in Mac OS X v10.1.

Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

NSApplication

Complete reference information is available in the NSApplication reference.

Instance Methods

cancelUserAttentionRequest:	Cancels a previous user attention request.
requestUserAttention:	Starts a user attention request.

Delegate Methods

applicationDockMenu:	Allows the delegate to supply a dock menu for the application dynamically.

NSAttributedString

Complete reference information is available in the NSAttributedString reference.

Class Methods

textFileTypes	Returns an array of strings representing those file types that can be loaded as text.
textPasteboardTypes	Returns an array of pasteboard types that can be loaded as text.
textUnfilteredFileTypes	Returns an array of strings representing those file types that can be loaded as a text.

textUnfilteredPasteboardTypes	Returns an array of pasteboard types that can be loaded as text.

NSCell

Complete reference information is available in the NSCell reference.

Instance Methods

highlightColorWithFrame:inView:	Returns the color the receiver uses when drawing the selection highlight.

NSColor

Complete reference information is available in the NSColor reference.

Class Methods

secondarySelectedControlColor	Returns the system color used in non-key views.

NSDocument

Complete reference information is available in the NSDocument reference.

Instance Methods

fileAttributesToWriteToFile:ofType:saveOperation:	Returns the file attributes that should be written to the named document file of the specified type.
fileNameExtensionWasHiddenInLastRunSavePanel	Returns YES if a Save panel was presented by this document and the user chose to hide the name extension of the file that was selected in that Save panel.

NSEvent

Complete reference information is available in the NSEvent reference.

Instance Methods

`buttonNumber`	Returns the button number for the mouse button that generated an NSOtherMouse... event.

NSObject

Complete reference information is available in the `NSObject` reference.

Instance Methods

`windowDidEndSheet:`	Sent by the default notification center immediately after an NSWindow object closes a sheet.
`windowWillBeginSheet:`	Sent by the default notification center immediately before an NSWindow object opens a sheet.

NSOpenPanel

Complete reference information is available in the `NSOpenPanel` reference.

Instance Methods

`resolvesAliases`	Returns whether the receiver resolves aliases.
`setResolvesAliases:`	Sets whether the receiver resolves aliases to resolvesAliases.

NSResponder

Complete reference information is available in the `NSResponder` reference.

Instance Methods

`otherMouseDown:`	Informs the receiver that the user has pressed a mouse button other than the left or right one.
`otherMouseDragged:`	Informs the receiver that the user has moved the mouse with a button other than the left or right button pressed.
`otherMouseUp:`	Informs the receiver that the user has released a mouse button other than the left or right button.

NSSavePanel

Complete reference information is available in the `NSSavePanel` reference.

Instance Methods

`isExtensionHidden`	Returns a Boolean value that indicates whether the extension-hiding checkbox is visible and checked.
`setCanSelectHiddenExtension:`	Sets whether the panel allows the user to hide or show extensions.
`setExtensionHidden:`	Sets the value of the extension-hiding checkbox.

NSTextView

Complete reference information is available in the `NSTextView` reference.

Instance Methods

`startSpeaking:`	Speaks the selected text, or all text if no selection.
`stopSpeaking:`	Stops the speaking of text.

NSView

Complete reference information is available in the `NSView` reference.

Instance Methods

`inLiveResize`	A convenience method, expected to be called from drawRect:, to assist in decisions about optimized drawing.
`setKeyboardFocusRingNeedsDisplayInRect:`	Invalidates the area around the focus ring.
`viewDidEndLiveResize`	Informs the receiver of the end of a live resize.
`viewWillStartLiveResize`	Informs the receiver of the start of a live resize.

NSWindow

Complete reference information is available in the `NSWindow` reference.

Instance Methods

attachedSheet	Returns the sheet attached to the window.
isSheet	Indicates whether the window has ever run as a modal sheet.

C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

NSApplication.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSAppKitVersionNumber	This value corresponds to one of the constants defined in "Application Kit framework version numbers."
NSAppKitVersionNumber10_0	The Application Kit framework included in Mac OS X v10.0.
NSCriticalRequest	The user attention request is a critical request.
NSInformationalRequest	The user attention request is an informational request.
NSRequestUserAttentionType	These constants specify the level of severity of a user attention request and are used by cancelUserAttentionRequest: and requestUserAttention:.

NSEvent.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSOtherMouseDown	See "Mouse Events".
NSOtherMouseDownMask	Corresponds to NSOtherMouseDown. See "Mouse Events".
NSOtherMouseDragged	See "Mouse Events".
NSOtherMouseDraggedMask	Corresponds to NSOtherMouseDragged. See "Mouse Events".

NSOtherMouseUp	See “Mouse Events”.
NSOtherMouseUpMask	Corresponds to NSOtherMouseUp. See “Mouse Events”.

NSGraphics.h

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSSetFocusRingStyle	Specifies how a focus ring will be drawn.

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSFocusRingAbove	Use NSFocusRingAbove to draw over an image.
NSFocusRingBelow	Use NSFocusRingBelow to draw the focus ring under text.
NSFocusRingOnly	Use NSFocusRingOnly if you don’t have an image or text.
NSFocusRingPlacement	The focus ring style indicates how the focus ring will be drawn.

NSPasteboard.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

NSHTMLPboardType	HTML (which an NSTextView object can read from, but not write to).

NSScroller.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSScrollerArrowsDefaultSetting`	Buttons are displayed according to the system-wide appearance preferences.

NSView.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSViewGlobalFrameDidChangeNotification`	Posted whenever an NSView object that has attached surfaces (that is, NSOpenGLContext objects) moves to a different screen, or other cases where the NSOpenGLContext object needs to be updated.

NSWindow.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`NSWindowDidEndSheetNotification`	Posted whenever an NSWindow object closes an attached sheet.
`NSWindowWillBeginSheetNotification`	Posted whenever an NSWindow object is about to open a sheet.

Document Revision History

This table describes the changes to *Application Kit Reference Update*.

Date	Notes
2007-07-18	Updated with the symbols added to the Application Kit framework in Mac OS X v10.5.
2006-05-23	Organized delegate methods and informal protocol methods.
2005-09-08	Added cross-reference to complete Application Kit reference to table of contents.
2005-08-11	Removed unused NSAccessibility constants.
2005-06-04	Entered editorial revisions throughout.
2005-04-29	New document that summarizes the symbols added to the Application Kit framework in Mac OS X v10.4.