NSCell

Inherits From: NSObject

Declared In: appkit/NSCell.h

Initializing an NSCell

- (id)**initImageCell:**(NSImage *)*anImage* Initializes a new NSCell with the NSImage *anImage.*
- (id)**initTextCell:**(NSString *)*aString* Initializes a new NSCell with title *aString*.

Determining Component Sizes

- (void)**calcDrawInfo:**(NSRect)*aRect* Implemented by subclasses to recalculate drawing sizes.
- (NSSize)**cellSize** Returns the minimum size needed to display the NSCell.
- (NSSize)**cellSizeForBounds:**(NSRect)*aRect* Returns the minimum size needed to display the NSCell.
- (NSRect)**drawingRectForBounds:**(NSRect)*theRect* Returns the rectangle the NSCell draws in.
- (NSRect)**imageRectForBounds:**(NSRect)*theRect* Returns the rectangle that the cell's image is drawn in.
- (NSRect)**titleRectForBounds:**(NSRect)*theRect* Returns the rectangle that the cell's title is drawn in.

Setting the NSCell's Type

- (void)**setType:**(NSCellType)*aType* Sets the NSCell's type to *aType*.
- (NSCellType)**type** Returns the NSCell's type.

Setting the NSCell's State

- (void)**setState:**(int)*value* Sets the state of the NSCell to *value* (0 or 1).
- (int)**state** Returns the state of the NSCell (0 or 1).

Enabling and Disabling the NSCell

- (BOOL)**isEnabled**	Returns whether the NSCell reacts to mouse events.
- (void)**setEnabled:**(BOOL)*flag*	Sets whether the NSCell reacts to mouse events.

Setting the Image

- (NSImage *)**image**	Returns the NSCell's image.
- (void)**setImage:**(NSImage *)*anImage*	Makes *anImage* the NSCell's image.

Setting the NSCell's Value

- (double)**doubleValue**	Returns the NSCell's value as a **double**.
- (float)**floatValue**	Returns the NSCell's value as a **float**.
- (int)**intValue**	Returns the NSCell's value as an **int**.
- (NSString *)**stringValue**	Returns the NSCell's value as a string.
- (void)**setDoubleValue:**(double)*aDouble*	Sets the NSCell's value to *aDouble*.
- (void)**setFloatValue:**(float)*aFloat*	Sets the NSCell's value to *aFloat*.
- (void)**setIntValue:**(int)*anInt*	Sets the NSCell's value to *anInt*.
- (void)**setStringValue:**(NSString *)*aString*	Sets the NSCell's value to a copy of *aString*.

Interacting with Other NSCells

- (void)**takeDoubleValueFrom:**(id)*sender*	Sets the NSCell's value to *sender*'s double floating-point value.
- (void)**takeFloatValueFrom:**(id)*sender*	Sets the NSCell's value to *sender*'s floating-point value.
- (void)**takeIntValueFrom:**(id)*sender*	Sets the NSCell's value to *sender*'s integer value.
- (void)**takeStringValueFrom:**(id)*sender*	Sets the NSCell's value to *sender*'s string value.

Modifying Text Attributes

- (NSTextAlignment)**alignment**	Returns the alignment of text in the NSCell.
- (NSFont *)**font**	Returns the Font used to display text in the NSCell.
- (BOOL)**isEditable**	Returns whether the NSCell's text is editable.
- (BOOL)**isSelectable**	Returns whether the NSCell's text is selectable.
- (BOOL)**isScrollable**	Returns whether the NSCell scrolls to follow typing.

- (void)**setAlignment:**(NSTextAlignment)*mode* Sets the alignment of text in the NSCell to *mode*.
- (void)**setEditable:**(BOOL)*flag* Sets whether the NSCell's text is editable.
- (void)**setFont:**(NSFont *)*fontObject* Sets the Font used to display text in the NSCell to *fontObject*.
- (void)**setSelectable:**(BOOL)*flag* Sets whether the NSCell's text is selectable.
- (void)**setScrollable:**(BOOL)*flag* Sets whether the NSCell scrolls to follow typing.
- (id)**setUpFieldEditorAttributes:**(id)*textObject* Sets NSText parameters for the field editor. (See the documentation for NSText.)
- (void)**setWrap:**(BOOL)*flag* Sets whether the NSCell's text is word-wrapped.

Editing Text

- (void)**editWithFrame:**(NSRect)*aRect* Allows text editing in response to a mouse-down event.
  inView:(NSView *)*controlView*
  editor:(NSText *)*textObject*
  delegate:(id)*anObject*
  event:(NSEvent *)*theEvent*
- (void)**endEditing:**(NSText *)*textObject* Ends any text editing occurring in the NSCell.
- (void)**selectWithFrame:**(NSRect)*aRect* Allows text selection in response to a mouse-down event.
  inView:(NSView *)*controlView*
  editor:(NSText *)*textObject*
  delegate:(id)*anObject*
  start:(int)*selStart*
  length:(int)*selLength*

Validating Input

- (int)**entryType** Returns the type of data the user can type into the NSCell.
- (BOOL)**isEntryAcceptable:**(NSString *)*aString* Returns whether *aString* is acceptable for the entry type.
- (void)**setEntryType:**(int)*aType* Sets the type of data the user can type into the NSCell.

Formatting Data

- (void)**setFloatingPointFormat:**(BOOL)*autoRange* Sets the display format for floating-point values.
  left:(unsigned int)*leftDigits*
  right:(unsigned int)*rightDigits*

Modifying Graphic Attributes

- (BOOL)**isBezeled**	Returns whether the NSCell has a bezeled border.
- (BOOL)**isBordered**	Returns whether NSCell has a plain border.
- (BOOL)**isOpaque**	Returns whether the NSCell is opaque.
- (void)**setBezeled:**(BOOL)*flag*	Sets whether the NSCell has a bezeled border.
- (void)**setBordered:**(BOOL)*flag*	Sets whether the NSCell has a plain border.

Setting Parameters

- (int)**cellAttribute:**(NSCellAttribute)*aParameter*	Returns various flag values.
- (void)**setCellAttribute:**(NSCellAttribute)*aParameter* **to:**(int)*value*	Sets various NSCell flags.

Displaying

- (NSControl *)**controlView**	Implemented by subclasses to return the NSControl last drawn in.
- (void)**drawInteriorWithFrame:**(NSRect)*cellFrame* **inView:**(NSView *)*controlView*	Draws the area within the NSCell's border in *controlView.*
- (void)**drawWithFrame:**(NSRect)*cellFrame* **inView:**(NSView *)*controlView*	Draws the entire NSCell in *controlView.*
- (void)**highlight:**(BOOL)*lit* **withFrame:**(NSRect)*cellFrame* **inView:**(NSView *)*controlView*	If *lit* is YES, highlights the NSCell in *controlView*, otherwise unhighlights.
- (BOOL)**isHighlighted**	Returns whether the NSCell is highlighted.

Target and Action

- (SEL)**action**	Implemented by subclasses to return the action method.
- (BOOL)**isContinuous**	Returns whether the NSCell continuously sends the action.
- (int)**sendActionOn:**(int)*mask*	Determines when the action is sent while tracking.
- (void)**setAction:**(SEL)*aSelector*	Implemented by subclasses to set the action method.
- (void)**setContinuous:**(BOOL)*flag*	Sets whether the NSCell continuously sends the action.
- (void)**setTarget:**(id)*anObject*	Implemented by subclasses to set the target object.
- (id)**target**	Implemented by subclasses to return the target object.

Assigning a Tag

- (void)**setTag:**(int)*anInt*	Implemented by subclasses to set an identifier tag.
- (int)**tag**	Implemented by subclasses to return the identifier tag.

Handling Keyboard Alternatives

- (NSString *)**keyEquivalent**	Implemented by subclasses to return a key equivalent.

Tracking the Mouse

+ (BOOL)**prefersTrackingUntilMouseUp**	Returns NO, so tracking stops when the mouse leaves the NSCell; subclasses may override.
- (BOOL)**continueTracking:**(NSPoint)*lastPoint* **at:**(NSPoint)*currentPoint* **inView:**(NSView *)*controlView*	Returns whether tracking should continue based on *lastPoint* and *currentPoint* within *controlView*.
- (int)**mouseDownFlags**	Returns the event flags set at the start of mouse tracking.
- (void)**getPeriodicDelay:**(float *)*delay* **interval:**(float *)*interval*	Returns repeat values for continuous sending of the action.
- (BOOL)**startTrackingAt:**(NSPoint)*startPoint* **inView:**(NSView *)*controlView*	Determines whether tracking should begin based on *startPoint* within *controlView*.
- (void)**stopTracking:**(NSPoint)*lastPoint* **at:**(NSPoint)*stopPoint* **inView:**(NSView *)*controlView* **mouseIsUp:**(BOOL)*flag*	Allows the NSCell to update itself to end tracking, based on *lastPoint* and *stopPoint* within *controlView*; flag is YES if this method was invoked because the mouse went up.
- (BOOL)**trackMouse:**(NSEvent *)*theEvent* **inRect:**(NSRect)*cellFrame* **ofView:**(NSView *)*controlView* **untilMouseUp:**(BOOL)*flag*	Tracks the mouse, returning YES if the mouse goes up while in *cellFrame*. This method is usually invoked by an NSControl's **mouseDown:** method, which passes the mouse-down event in *theEvent*. If *flag* is YES, the method keeps tracking until the mouse goes up; otherwise it tracks until the mouse leaves *cellFrame.*

Managing the Cursor

- (void)**resetCursorRect:**(NSRect)*cellFrame*	Sets text NSCells to show the I-beam cursor.

inView:(NSView *)*controlView*

Comparing to Another NSCell

- (NSComparisonResult)**compare**:(id)*otherCell* — Compares the string values of this cell and *otherCell* (which must be a kind of NSCell).

Using the NSCell to Represent an Object

- (id)**representedObject** — Returns the object that the receiver represents, if any.
- (void)**setRepresentedObject:**(id)*anObject* — Creates an association between the receiver and *anObject*. *anObject* will be retained, released, archived, and unarchived whenever the receiver is. If another cell is already associated with *anObject,* that association is broken, and the receiver is associated with the object.