NSCStringText

Inherits From: NSText : NSView : NSResponder : NSObject

Conforms To: NSChangeSpelling
                             NSIgnoreMisspelledWords

Declared In: appkit/NSCStringText.h

Initializing a New NSCStringText Object

- (id)**initWithFrame:**(NSRect)*frameRect*   Returns a new NSCStringText object at *frameRect*
     text:(NSString *)*theText*   initialized with the contents of *theText* and with *mode*
     alignment:(NSTextAlignment)*mode*   alignment.

Modifying the Frame Rectangle

- (void)**resizeTextWithOldBounds:**(NSRect)*oldBounds*
     maxRect:(NSRect)*maxRect*   Used by the NSCStringText object to resize and redisplay itself.

Laying Out the Text

- (int)**calcLine**   Calculates line breaks.
- (BOOL)**changeTabStopAt:**(float)*oldX*   Resets the position of the specified tab stop.

to:(float)*newX*
- (BOOL)**charWrap**   Returns whether extra long words are wrapped.
- (void *)**defaultParagraphStyle**   Returns the default paragraph style.
- (float)**descentLine**   Returns distance from base line to bottom of line.
- (void)**getMarginLeft:**(float *)*leftMargin*   Gets by reference the dimensions of margins around the
  right:(float *)*rightMargin*   text.
  top:(float *)*topMargin*
  bottom:(float *)*bottomMargin*
- (void)**getMinWidth:**(float *)*width*   Given the *widthMax* and *heightMax*, calculates the
  minHeight:(float *)*height*   minimum area needed to display the text and returns
  maxWidth:(float)*widthMax*   *width* and *height* by reference.
  maxHeight:(float)*heightMax*
- (float)**lineHeight**   Returns height of a line of text.
- (void *)**paragraphStyleForFont:**(NSFont *)*fontId*   Recalculates the paragraph style based on new font *fontId*
  alignment:(int)*alignment*   and *alignment*.
- (void)**setCharWrap:**(BOOL)*flag*   Sets whether extra long words are wrapped.
- (void)**setDescentLine:**(float)*value*   Sets the distance from the   base line to the bottom of line to *value*.
- (void)**setLineHeight:**(float)*value*   Sets the height of a line of text to *value*.
- (void)**setMarginLeft:**(float)*leftMargin*   Adjusts the margins around the text.
  right:(float)*rightMargin*
  top:(float)*topMargin*
  bottom:(float)*bottomMargin*
- (void)**setNoWrap**   Disables word wrap.
- (void)**setParagraphStyle:**(void *)*paraStyle*   Sets the default paragraph style for the entire text.
- (BOOL)**setSelProp:**(NSParagraphProperty)*property*
  to:(float)*value*   Sets a paragraph *property* for one or more selected
                                                          paragraphs to *value*.

Reporting Line and Position

- (int)**lineFromPosition:**(int)*position*   Converts character *position* to line number.

- (int)**positionFromLine:**(int)*line*   Converts *line* number to character position.

Reading and Writing Text

- (void)**finishReadingRichText**   Sent after the NSCStringText object reads RTF data.
- (NSTextBlock *)**firstTextBlock**   Returns a pointer to the first text block in the NSCStringText object.
- (NSRect)**paragraphRect:**(int)*paraNumber*   Returns the location and size of a paragraph identified by
    start:(int *)*startPos*   *paraNumber*;   also returns the starting and ending
    end:(int *)*endPos*   character positions by reference.
- (void)**startReadingRichText**   Sent before the NSCStringText object begins reading RTF data.

Editing Text

- (void)**clear:**(id)*sender*   Deletes the selected text.
- (void)**hideCaret**   Removes the caret from the text display.
- (void)**showCaret**   Displays the previously hidden caret in the text display.

Managing the Selection

- (void)**getSelectionStart:**(NSSelPt *)*start*   Gets information (by reference) relating to the starting and
    end:(NSSelPt *)*end*   ending character positions of the selection.
- (void)**replaceSel:**(NSString *)*aString*   Replaces the selection with *aString*.
- (void)**replaceSel:**(NSString *)*aString*   Replaces the selection with *length* bytes of *aString*.
    length:(int)*length*

- (void)**replaceSel:**(NSString *)*aString*   Replaces the selection with *length* bytes of *aString*.
    length:(int)*length*   *insertRuns* is a pointer to the current run in the run
    runs:(NSRunArray *)*insertRuns*   array.
- (void)**scrollSelToVisible**   Brings the selection within the frame rectangle.
- (void)**selectError**   Selects all the text.
- (void)**selectNull**   Deselects the current selection.

- (void)**setSelectionStart:**(int)*start*   Selects text from characters *start* through *end*.
      end:(int)*end*
- (void)**selectText:**(id)*sender*   Makes the receiver the first responder and selects all text.

Setting the Font

+ (NSFont *)**defaultFont**   Returns the default NSFont object for NSCStringText objects.
+ (void)**setDefaultFont:**(NSFont *)*anObject*   Makes *anObject* the default NSFont object for NSCStringText objects.
- (void)**setFont:**(NSFont *)*fontObj*   Sets the NSFont object and paragraph style for all text.
      paragraphStyle:(void *)*paragraphStyle*
- (void)**setSelFont:**(NSFont *)*fontObj*   Sets the NSFont object for the selection.
- (void)**setSelFont:**(NSFont *)*fontObj*   Sets the NSFont object and paragraph style for the
      paragraphStyle:(void *)*paragraphStyle*   selection.
- (void)**setSelFontFamily:**(NSString *)*fontName* Sets the font family for the selection.
- (void)**setSelFontSize:**(float)*size*   Sets the font size for the selection.
- (void)**setSelFontStyle:**(NSFontTraitMask)*traits* Sets the font style for the selection.

Finding Text

- (BOOL)**findText:**(NSString *)*textPattern*   Searches for *textPattern* in the text, starting at the insertion
      ignoreCase:(BOOL)*ignoreCase*   point.   *ignoreCase* instructs the search to disregard case;
      backwards:(BOOL)*backwards*   *backwards* means search backwards;   *wrap* means that when the
                                                search reaches the beginning or end of the text (depending on the
                                              direction), it should continue by wrapping to the end or beginning of the
                                              text.

Modifying Graphic Attributes

- (NSColor *)**runColor:**(NSRun *)*run*   Returns the color of the specified text run.
- (NSColor *)**selColor**   Returns the color of the selected text.

- (void)**setSelColor:**(NSColor *)*color*   Sets the color of the selected text.

Reusing an NSCStringText Object

- (void)**renewFont:**(NSFont *)*newFontObj*   Resets the NSCStringText object to draw different text
    text:(NSString *)*newText*   *newText* in font *newFontId* within frame *newFrame*.
    frame:(NSRect)*newFrame*
    tag:(int)*newTag*
- (void)**renewFont:**(NSString *)*newFontName*   Resets the NSCStringText object to draw different text
    size:(float)*newFontSize*   *newText* in the font identified by *newFontName*,
    style:(int)*newFontStyle*   *newFontSize*, and *newFontStyle*. Drawing occurs within
    text:(NSString *)*newText*   frame *newFrame*.
    frame:(NSRect)*newFrame*
    tag:(int)*newTag*
- (void)**renewRuns:**(NSRunArray *)*newRuns*   Resets the NSCStringText object to draw different text
    text:(NSString *)*newText*   *newText* in *newFrame*.
    frame:(NSRect)*newFrame*
    tag:(int)*newTag*
- (void)**windowChanged:**(NSWindow *)*newWindow*   Hides the caret whenever the NSCStringText's window changes.

Setting Window Attributes

- (BOOL)**isRetainedWhileDrawing**   Returns whether a retained window is used for drawing.
- (void)**setRetainedWhileDrawing:**(BOOL)*flag*   Allows use of a retained window when drawing.

Assigning a Tag

- (void)**setTag:**(int)*anInt*   Makes *anInt* the NSCStringText object's tag.
- (int)**tag**   Returns the NSCStringText object's tag.

Handling Event Messages

- (void)**becomeKeyWindow**   Activates the caret if selection has width of 0.
- (void)**moveCaret:**(unsigned short)*theKey*   Moves the caret in response to arrow keys.
- (void)**resignKeyWindow**   Deactivates the caret.

Displaying Graphics within the Text

- \+ **registerDirective:**(NSString *)*directive*   Associates an RTF control word (*directive*) with   *class*
        forClass:*class*   (usually NSCell and subclasses);   objects of this class are encoded through RTF control words in NSCStringText objects.
- (NSPoint)**locationOfCell:**(NSCell *)*cell*   Returns the location of *cell*.
- (void)**replaceSelWithCell:**(NSCell *)*cell*   Replaces the selection with cell object *cell*.
- (void)**setLocation:**(NSPoint)*origin*   Sets the *origin* point of *cell*.
        ofCell:(NSCell *)*cell*

Using the Services Menu and the Pasteboard

- \+ **excludeFromServicesMenu:**(BOOL)*flag*   Controls whether NSCStringText objects can register for services.
- (BOOL)**readSelectionFromPasteboard:**(NSPasteboard *)*pboard*
                                         Replaces the selection with data from pasteboard *pboard*.
- (id)**validRequestorForSendType:**(NSString *)*sendType*
        returnType:(NSString *)*returnType*   Determines which Service menu items are enabled.
- (BOOL)**writeSelectionToPasteboard:**(NSPasteboard *)*pboard*
        types:(NSArray *)*types*   Copies the selection to pasteboard *pboard*.

Setting Tables and Functions

- (const NSFSM *)**breakTable**   Returns the table defining word boundaries.
- (const unsigned char *)**charCategoryTable**   Returns the table defining character categories.

- (NSCharFilterFunc)**charFilter**   Returns the current character filter function.
- (const NSFSM *)**clickTable**   Returns the table defining double-click selection.
- (NSTextFunc)**drawFunc**   Returns the current draw function.
- (const unsigned char *)**postSelSmartTable**   Returns cut and paste table for right word boundary.
- (const unsigned char *)**preSelSmartTable**   Returns cut and paste table for left word boundary.
- (NSTextFunc)**scanFunc**   Returns the current scan function.
- (void)**setBreakTable:**(const NSFSM *)*aTable*   Sets the table defining word boundaries.
- (void)**setCharCategoryTable:**(const unsigned char *)*aTable*

> Sets the table defining character categories used in the word wrap or click tables.

- (void)**setCharFilter:**(NSCharFilterFunc)*aFunction*

> Makes *aFunction* the character filter function.

- (void)**setClickTable:**(const NSFSM *)*aTable*   Sets the table defining double-click selection.
- (void)**setDrawFunc:**(NSTextFunc)*aFunction*   Makes *aFunction* the function that draws the text.
- (void)**setPostSelSmartTable:**(const unsigned char *)*aTable*

> Sets the cut and paste table for right word boundary.

- (void)**setPreSelSmartTable:**(const unsigned char *)*aTable*

> Sets the cut and paste table for left word boundary.

- (void)**setScanFunc:**(NSTextFunc)*aFunction*   Makes *aFunction* the scan function.
- (void)**setTextFilter:**(NSTextFilterFunc)*aFunction*Makes *aFunction* the text filter function.
- (NSTextFilterFunc)**textFilter**   Returns the current text filter function.

Printing

- (void)**adjustPageHeightNew:**(float *)*newBottom*   Assists with automatic pagination of text.
     top:(float)*oldTop*
     bottom:(float)*oldBottom*
     limit:(float)*bottomLimit*

Implemented by an Embedded Graphic Object

- (NSSize)**cellSize**   Embedded cell returns its size.
- (void)**drawWithFrame:**(NSRect)*cellFrame*   Embedded object draws itself, including frame, within
    inView:(NSView *)*controlView*   *cellFrame* in *controlView*.
- (void)**highlight:**(BOOL)*flag*
    withFrame:(NSRect)*cellFrame*   Embedded object highlights or unhighlights itself with
    inView:(NSView *)*controlView*   *cellFrame* of *controlView*, depending on the value of *flag*.
- (void)**readRichText:**(NSString *)*stringObject*   Embedded object reads its RTF representation from
    forView:(NSView *)*view*   *stringObject* and initializes itself.
- (NSString *)**richTextForView:**(NSView *)*view*   Embedded object stores its RTF representation within view as a string
                                                            object and returns it.
- (BOOL)**trackMouse:**(NSEvent *)*theEvent*   Embedded object implements this method to track mouse
    inRect:(NSRect)*cellFrame*   movement within tracking rectangle (*cellFrame*) and to
    ofView:(NSView *)*controlView*   detect mouse-up event (*untilMouseUp*).
    untilMouseUp:(BOOL)*untilMouseUp*

Implemented by the Delegate

- (void)**textDidRead:**(NSText *)*textObject*   Lets the delegate review paper size.
    paperSize:(NSSize)*paperSize*
- (NSRect)**textDidResize:**(NSText *)*textObject*   Reports size change to delegate.
    oldBounds:(NSRect)*oldBounds*
- (NSFont *)**textWillConvert:**(NSText *)*textObject*
    fromFont:(NSFont *)*font*   Lets delegate intercede in selection's font change.
    toFont:(NSFont *)*font*
- (void)**textWillFinishReadingRichText:**(NSText *)*textObject*
                                                            Informs delegate that the NSCStringText object finished reading RTF data.
- (void)**textWillResize:**(NSText *)*textObject*   Informs delegate of impending size change.
- (void)**textWillSetSel:**(NSText *)*textObject*   Lets delegate intercede in the updating of the Font panel.
    toFont:(NSFont *)*font*

\- (void)**textWillStartReadingRichText:**(NSText *)*textObject*

Informs delegate that NSCStringText object will read RTF data.

\- (NSSize)**textWillWrite:**(NSText *)*textObject*   Lets the delegate specify paper size.