NSTextAttachmentCell

Adopted By: NSTextAttachmentCell

Declared In: AppKit/NSTextAttachment.h

Protocol Description

The NSTextAttachmentCell protocol declares the interface for objects that draw text attachment icons and handle mouse events on their icons. With the exceptions of **cellBaselineOffset:**, **setAttachment:** and **attachment**, all of these methods are implemented by the NSCell class and described in that class specification.

See the NSAttributedString and NSTextView class specifications for general information on text attachments.

Method Types

Drawing	– drawWithFrame:inView: – highlight:withFrame:inView:
Cell size and position	– cellSize – cellBaselineOffset
Event handling	– wantsToTrackMouse – trackMouse:inRect:ofView:untilMouseUp:
Setting the attachment	– setAttachment: – attachment

Instance Methods

attachment

– (NSTextAttachment *)**attachment**

Returns the text attachment object that owns the receiver.

See also: – **setAttachment:**

cellBaselineOffset

– (NSPoint)**cellBaselineOffset**

Returns the position where the attachment cell’s image should be drawn in text, relative to the current point established in the glyph layout. The image should be drawn so that its lower left corner lies on this point.

See also: – **icon** (NSFileWrapper)

cellSize

– (NSSize)**cellSize**

Returns the size of the attachment’s icon.

See also: – **icon** (NSFileWrapper), – **fileWrapper** (NSTextAttachment)

drawWithFrame:inView:

– (void)**drawWithFrame:**(NSRect)*cellFrame* **inView:**(NSView *)*aView*

Draws the receiver’s image within *cellFrame* in *aView*, which should be the focus view.

See also: – **drawWithFrame:inView:** (NSCell), – **lockFocus** (NSView)

highlight:withFrame:inView:

– (void)**highlight:**(BOOL)*flag*
withFrame:(NSRect)*cellFrame*
inView:(NSView *)*aView*

Draws the receiver’s image—with highlighting if *flag* is YES—within *cellFrame* in *aView*, which should be the focus view.

See also: – **highlight:withFrame:inView:** (NSCell), – **lockFocus** (NSView)

setAttachment:

– (void)**setAttachment:**(NSTextAttachment *)*anAttachment*

Sets the text attachment object that owns the receiver to *anAttachment*, without retaining it (the text attachment, as the owner, retains the cell).

See also: – **attachment**, – **setAttachmentCell:** (NSTextAttachment)

trackMouse:inRect:ofView:untilMouseUp:

– (BOOL)**trackMouse:**(NSEvent *)*theEvent*
inRect:(NSRect)*cellFrame*
ofView:(NSView *)*aTextView*
untilMouseUp:(BOOL)*flag*

Handles a mouse-down event on the receiver’s image. *theEvent* is the mouse-down event. *cellFrame* is the region of *aTextView* in which further mouse events should be tracked. *aTextView* is the view which received the event. It’s assumed to be an NSTextView, and should be the focus view. If *flag* is YES, the receiver tracks the mouse until a mouse-up event occurs; if *flag* is NO, it stops tracking when a mouse-dragged event occurs outside of *cellFrame*. Returns YES if the receiver successfully finished tracking the mouse (typically through a mouse-up event), NO otherwise (such as when the mouse is dragged outside *cellFrame*).

NSTextAttachmentCell’s implementation of this method calls upon *aTextView*’s delegate to handle the event. If *theEvent* is a mouse-up event for a double click, the text attachment cell sends the delegate a **textView:doubleClickedOnCell:inRect:** message and returns YES. Otherwise, depending on whether the user clicks or drags the cell, it sends the delegate a **textView:clickedOnCell:inRect:** or a **textView:draggingCell:inRect:event:** message and returns YES. NSTextAttachmentCell’s implementation returns NO only if *flag* is NO and the mouse is dragged outside of *cellFrame*. The delegate methods are invoked only if the delegate responds.

See also: – **wantsToTrackMouse**, – **trackMouse:inRect:ofView:untilMouseUp:** (NSCell),
– **lockFocus** (NSView)

wantsToTrackMouse

– (BOOL)**wantsToTrackMouse**

Returns YES if the receiver will handle a mouse event occurring over its image (to support dragging, for example), NO otherwise. NSTextAttachmentCell’s implementation of this method returns YES. The NSView containing the cell should invoke this method before sending a **trackMouse:inRect:ofView:untilMouseUp:** message.

For an attachment in an attributed string, if the attachment cell returns NO its attachment character should be selected rather than the cell being asked to track the mouse. this results in the attachment icon behaving as any regular glyph in text.