NSTextInputClient Protocol Reference

Cocoa > Events & Other Input

2008-10-15

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

Simultaneously published in the United States and Canada.

Contents

NSTextInputClient Protocol Reference

Adopted by	NSTextView
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	NSTextInputClient.h

Overview

The `NSTextInputClient` protocol defines the methods that Cocoa text views must implement in order to interact properly with the text input management system. To create another text view class, you can either subclass `NSTextView` (and not `NSText`, for historical reasons), or subclass `NSView` and implement the `NSTextInputClient` protocol

Important: Methods specific to the `NSTextInputClient` protocol are intended for dealing with text input and generally are not suitable for other purposes.

Tasks

Handling Marked Text

- `hasMarkedText` (page 10)
  Returns a Boolean value indicating whether the receiver has marked text.
- `markedRange` (page 11)
  Returns the range of the marked text.
- `selectedRange` (page 11)
  Returns the range of selected text.
- `setMarkedText:selectedRange:replacementRange:` (page 12)
  Replaces a specified range in the receiver's text storage with the given string and sets the selection.
- `unmarkText` (page 12)
  Unmarks the marked text.
- `validAttributesForMarkedText` (page 13)
  Returns an array of attribute names recognized by the receiver.

Storing Text

- `attributedSubstringForProposedRange:actualRange:` (page 7)
  Returns an attributed string derived from the given range in the receiver's text storage.
- `insertText:replacementRange:` (page 10)
  Inserts the given string into the receiver, replacing the specified content.

Getting Character Coordinates

- `characterIndexForPoint:` (page 8)
  Returns the index of the character whose bounding rectangle includes the given point.
- `firstRectForCharacterRange:actualRange:` (page 9)
  Returns the first logical boundary rectangle for characters in the given range.

Binding Keystrokes

- `doCommandBySelector:` (page 8)
  Invokes the action specified by the given selector.

Optional Methods

- `attributedString` (page 6)
  Returns an attributed string representing the receiver's text storage. (optional)
- `fractionOfDistanceThroughGlyphForPoint:` (page 9)
  Returns the fraction of the distance from the left side of the character to the right side that a given point lies. (optional)
- `baselineDeltaForCharacterAtIndex:` (page 7)
  Returns the baseline position of a given character relative to the origin of rectangle returned by `firstRectForCharacterRange:actualRange:` (page 9). (optional)
- `windowLevel` (page 13)
  Returns the window level of the receiver. (optional)

Instance Methods

attributedString

Returns an attributed string representing the receiver's text storage. (optional)

```
- (NSAttributedString *)attributedString
```

Return Value
The attributed string of the receiver's text storage.

Discussion
Implementation of this method is optional. A class adopting the `NSTextInputClient` protocol can implement this interface if it can be done efficiently to enable callers of this interface to access arbitrary portions of the receiver's content more efficiently.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSTextInputClient.h`

attributedSubstringForProposedRange:actualRange:

Returns an attributed string derived from the given range in the receiver's text storage.

```
- (NSAttributedString *)attributedSubstringForProposedRange:(NSRange)aRange
    actualRange:(NSRangePointer)actualRange
```

Parameters

aRange
The range in the text storage from which to create the returned string.

actualRange
The actual range of the returned string if it was adjusted, for example, to a grapheme cluster boundary or for performance or other reasons. `NULL` if range was not adjusted.

Return Value
The string created from the given range. May return `nil`.

Discussion
An implementation of this method should be prepared for *aRange* to be out of bounds. For example, the InkWell text input service can ask for the contents of the text input client that extends beyond the document's range. In this case, you should return the intersection of the document's range and *aRange*. If the location of *aRange* is completely outside of the document's range, return `nil`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSTextInputClient.h`

baselineDeltaForCharacterAtIndex:

Returns the baseline position of a given character relative to the origin of rectangle returned by `firstRectForCharacterRange:actualRange:` (page 9). (optional)

```
- (CGFloat)baselineDeltaForCharacterAtIndex:(NSUInteger)anIndex
```

Parameters

anIndex
Index of the character whose baseline is tested.

Return Value
The vertical distance, in points, between the baseline of the character at *anIndex* and the rectangle origin.

Discussion
Implementation of this method is optional. This information allows the caller to determine finer-grained character positioning within the text storage of the text view adopting `NSTextInputClient`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSTextInputClient.h`

characterIndexForPoint:

Returns the index of the character whose bounding rectangle includes the given point.

```
- (NSUInteger)characterIndexForPoint:(NSPoint)aPoint
```

Parameters

aPoint

The point to test, in screen coordinates.

Return Value
The character index, measured from the start of the receiver's text storage, of the character containing the given point. Returns `NSNotFound` if the cursor is not within a character's bounding rectangle.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSTextInputClient.h`

doCommandBySelector:

Invokes the action specified by the given selector.

```
- (void)doCommandBySelector:(SEL)aSelector
```

Parameters

aSelector

The selector to invoke.

Discussion
If *aSelector* cannot be invoked, then `doCommandBySelector:` should not pass this message up the responder chain. `NSResponder` also implements this method, and it does forward uninvokable commands up the responder chain, but a text view should not. A text view implementing the `NSTextInputClient` protocol inherits from `NSView`, which inherits from `NSResponder`, so your implementation of this method will override the one in `NSResponder`. It should not call `super`.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `interpretKeyEvents:` (NSResponder)
- `doCommandBySelector:` (NSResponder)

Declared In
`NSTextInputClient.h`

firstRectForCharacterRange:actualRange:

Returns the first logical boundary rectangle for characters in the given range.

```
- (NSRect)firstRectForCharacterRange:(NSRange)aRange
    actualRange:(NSRangePointer)actualRange
```

Parameters

aRange

The character range whose boundary rectangle is returned.

actualRange

If non-`NULL`, contains the character range corresponding to the returned area if it was adjusted, for example, to a grapheme cluster boundary or characters in the first line fragment.

Return Value
The boundary rectangle for the given range of characters, in screen coordinates. The rectangle's `size` value can be negative if the text flows to the left.

Discussion
If *aRange* spans multiple lines of text in the text view, the rectangle returned is the one surrounding the characters in the first line. In that case *actualRange* contains the range covered by the first rect, so you can query all line fragments by invoking this method repeatedly. If the length of *aRange* is 0 (as it would be if there is nothing selected at the insertion point), the rectangle coincides with the insertion point, and its width is 0.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSTextInputClient.h`

fractionOfDistanceThroughGlyphForPoint:

Returns the fraction of the distance from the left side of the character to the right side that a given point lies. (optional)

```
- (CGFloat)fractionOfDistanceThroughGlyphForPoint:(NSPoint)aPoint
```

Parameters

aPoint

The point to test.

Return Value
The fraction of the distance *aPoint* is through the glyph in which it lies. May be 0 or 1 if *aPoint* is not within the bounding rectangle of a glyph (0 if the point is to the left or above the glyph; 1 if it's to the right or below).

Discussion
Implementation of this method is optional. This allows caller to perform precise selection handling.

For purposes such as dragging out a selection or placing the insertion point, a partial percentage less than or equal to 0.5 indicates that *aPoint* should be considered as falling before the glyph; a partial percentage greater than 0.5 indicates that it should be considered as falling after the glyph. If the nearest glyph doesn't lie under *aPoint* at all (for example, if *aPoint* is beyond the beginning or end of a line), this ratio is 0 or 1.

For example, if the glyph stream contains the glyphs "A" and "b", with the width of "A" being 13 points, and *aPoint* is 8 points from the left side of "A", then the fraction of the distance is 8/13, or 0.615. In this case, the *aPoint* should be considered as falling between "A" and "b" for purposes such as dragging out a selection or placing the insertion point.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSTextInputClient.h`

hasMarkedText

Returns a Boolean value indicating whether the receiver has marked text.

```
- (BOOL)hasMarkedText
```

Return Value
`YES` if the receiver has marked text; otherwise `NO`.

Discussion
The text view itself may call this method to determine whether there currently is marked text. `NSTextView`, for example, disables the Edit > Copy menu item when this method returns `YES`.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- markedRange` (page 11)

Declared In
`NSTextInputClient.h`

insertText:replacementRange:

Inserts the given string into the receiver, replacing the specified content.

```
- (void)insertText:(id)aString
    replacementRange:(NSRange)replacementRange
```

Parameters

aString

The text to insert, either an `NSString` or `NSAttributedString` instance.

replacementRange

The range of content to replace in the receiver's text storage.

Discussion
This method is the entry point for inserting text typed by the user and is generally not suitable for other purposes. Programmatic modification of the text is best done by operating on the text storage directly. Because this method pertains to the actions of the user, the text view must be editable for the insertion to work.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSTextInputClient.h`

markedRange

Returns the range of the marked text.

```
- (NSRange)markedRange
```

Return Value
The range of marked text or `{NSNotFound, 0}` if there is no marked range.

Discussion
The returned range measures from the start of the receiver's text storage. The return value's `location` is `NSNotFound` and its `length` is 0 if and only if `hasMarkedText` (page 10) returns `NO`.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `hasMarkedText` (page 10)
- `setMarkedText:selectedRange:replacementRange:` (page 12)
- `unmarkText` (page 12)

Declared In
`NSTextInputClient.h`

selectedRange

Returns the range of selected text.

```
- (NSRange)selectedRange
```

Return Value
The range of selected text or `{NSNotFound, 0}` if there is no selection.

Discussion
The returned range measures from the start of the receiver's text storage, that is, from 0 to the document length.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `setMarkedText:selectedRange:replacementRange:` (page 12)

Declared In
`NSTextInputClient.h`

setMarkedText:selectedRange:replacementRange:

Replaces a specified range in the receiver's text storage with the given string and sets the selection.

```
- (void)setMarkedText:(id)aString
    selectedRange:(NSRange)selectedRange
    replacementRange:(NSRange)replacementRange
```

Parameters

aString
: The string to insert. Can be either an `NSString` or `NSAttributedString` instance.

selectedRange
: The range to set as the selection, computed from the beginning of the inserted string.

replacementRange
: The range to replace, computed from the beginning of the marked text.

Discussion
If there is no marked text, the current selection is replaced. If there is no selection, the string is inserted at the insertion point.

When *aString* is an `NSString` object, the receiver is expected to render the marked text with distinguishing appearance (for example, `NSTextView` renders with `markedTextAttributes`).

Availability
Available in Mac OS X v10.5 and later.

See Also
- `selectedRange` (page 11)
- `unmarkText` (page 12)

Declared In
`NSTextInputClient.h`

unmarkText

Unmarks the marked text.

```
- (void)unmarkText
```

Discussion
The receiver removes any marking from pending input text and disposes of the marked text as it wishes. The text view should accept the marked text as if it had been inserted normally. If there is no marked text, the invocation of this method has no effect.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `selectedRange` (page 11)
- `setMarkedText:selectedRange:replacementRange:` (page 12)

Declared In
`NSTextInputClient.h`

validAttributesForMarkedText

Returns an array of attribute names recognized by the receiver.

```
- (NSArray*)validAttributesForMarkedText
```

Return Value
An array of `NSString` objects representing names for the supported attributes.

Discussion
Returns an empty array if no attributes are supported. See *NSAttributedString Application Kit Additions Reference* for the set of string constants representing standard attributes.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSTextInputClient.h`

windowLevel

Returns the window level of the receiver. (optional)

```
- (NSInteger)windowLevel
```

Return Value
The window level of the receiver.

Discussion
Implementation of this method is optional. A class adopting `NSTextInputClient` can implement this interface to specify its window level if it is higher than `NSFloatingWindowLevel`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSTextInputClient.h`

Document Revision History

This table describes the changes to *NSTextInputClient Protocol Reference*.

Date	Notes
2008-10-15	New document that describes the protocol enabling conforming objects to receive keyboard input as properly formed text.

Document Revision History

Index