Documentation

Language: **Swift**

AppKit / NSTextInputClient

Protocol

NSTextInputClient

A set of methods that text views need to implement to interact properly with the text input management system.

macOS

```
protocol NSTextInputClient
```

Mentioned in

Adopting the system text cursor in custom text views

Overview

To create another text view class, you can either subclass `NSTextView`, or subclass `NSView` and implement the `NSTextInputClient` protocol.

> **Important**
>
> Methods specific to the `NSTextInputClient` protocol are intended for dealing with text input and generally aren't suitable for other purposes.

Topics

Handling marked text

```
func hasMarkedText() -> Bool
```

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

Required

```
func markedRange() -> NSRange
```

Returns the range of the marked text.

Required

```
func selectedRange() -> NSRange
```

Returns the range of selected text.

Required

```
func setMarkedText(Any, selectedRange: NSRange, replacementRange: NSRange)
```

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

Required

```
func unmarkText()
```

Unmarks the marked text.

Required

```
func validAttributesForMarkedText() -> [NSAttributedString.Key]
```

Returns an array of attribute names recognized by the receiver.

Required

Storing text

```
func attributedString() -> NSAttributedString
```

Returns an attributed string representing the receiver’s text storage.

```
func attributedSubstring(forProposedRange: NSRange, actualRange: NSRangePointer?) -> NSAttributedString?
```

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

Required

```
func insertText(Any, replacementRange: NSRange)
```

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

Required

Getting character coordinates

```
func characterIndex(for: NSPoint) -> Int
```

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

Required

```
func firstRect(forCharacterRange: NSRange, actualRange: NSRangePointer?) -> NSRect
```

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

Required

```
func baselineDeltaForCharacter(at: Int) -> CGFloat
```

Returns the baseline position of a given character relative to the origin of rectangle returned by `firstRect(forCharacterRange:actualRange:)`.

```
func drawsVerticallyForCharacter(at: Int) -> Bool
```

Informs the text input management system whether the protocol-conforming client renders the character at the given index vertically.

```
func fractionOfDistanceThroughGlyph(for: NSPoint) -> CGFloat
```

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

Placing content

```
var documentVisibleRect: NSRect
```

```
var unionRectInVisibleSelectedRange: NSRect
```

```
func preferredTextAccessoryPlacement() -> NSTextCursorAccessoryPlacement
```

```
func windowLevel() -> Int
```

Returns the window level of the receiver.

Binding keystrokes

```
func doCommand(by: Selector)
```

Invokes the action specified by the given selector.

Required

Supporting adaptive images

```
var supportsAdaptiveImageGlyph: Bool
```

A Boolean value that indicates whether the document supports adaptive images in the input.

`func insert(NSAdaptiveImageGlyph, replacementRange: NSRange)`

Inserts an adaptive image into the text at the specifed location.

Relationships

Inherited By

`NSTextCheckingClient`

Conforming Types

`NSTextView`

See Also

Text input

Adopting the system text cursor in custom text views

Incorporate the system text cursor into your custom text UI in AppKit.

`class NSTextInputContext`

An object that represents the Cocoa text input system.

`class NSTextAlternatives`

A list of alternative strings for a piece of text.

`protocol NSTextContent`

A protocol that describes specific kinds of input content types.

`class NSTextInsertionIndicator`

A view that represents the insertion indicator in text.

`enum DisplayMode`

Constants that determine how to display the system text cursor in a custom text UI.

`struct AutomaticModeOptions`

Options that affect the automatic display mode.