
IMKTextInput Protocol Reference

[Cocoa > Internationalization](#)



2007-06-05



Apple Inc.
© 2007 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

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

Apple, the Apple logo, Carbon, Cocoa, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY

DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

IMKTextInput Protocol Reference 5

Overview	5
Tasks	5
Working with Text Ranges	5
Working with Character Indexes	5
Getting Attributes and Length	6
Managing Text Input	6
Instance Methods	6
attributedStringFromRange:	6
attributesForCharacterIndex:lineHeightRectangle:	7
bundleIdentifier	7
characterIndexForPoint:tracking:inMarkedRange:	8
insertText:replacementRange:	8
length	9
markedRange	9
overrideKeyboardWithKeyboardNamed:	10
selectedRange	10
selectInputMode:	10
setMarkedText:selectionRange:replacementRange:	11
supportsUnicode	11
validAttributesForMarkedText	12
windowLevel	12
Constants	13
IMKLocationToOffsetMappingMode	13
IMKTextOrientationName	13

Document Revision History 15

Index 17

IMKTextInput Protocol Reference

Adopted by	IMKInputSession
Framework	System/Library/Frameworks/Carbon.framework/HIToolbox.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	IMKInputSession.h

Overview

The `IMKTextInput` protocol defines methods for communicating with client input sessions. An input method uses this protocol to send text or obtain information about client text.

Tasks

Working with Text Ranges

- [insertText:replacementRange:](#) (page 8)
Sends fully converted text to an input session.
- [setMarkedText:selectionRange:replacementRange:](#) (page 11)
Inserts the provided text and marks it to indicate that it is part of an active input session.
- [selectedRange](#) (page 10)
Returns the current selection range.
- [markedRange](#) (page 9)
Returns the range in the document that is occupied by marked text, that is, the current inline session.
- [attributedStringFromRange:](#) (page 6)
Returns the attributed string for the provided range of text.

Working with Character Indexes

- [characterIndexForPoint:tracking:inMarkedRange:](#) (page 8)
Returns the location in the text document that maps to a global point.
- [attributesForCharacterIndex:lineHeightRectangle:](#) (page 7)
Returns a dictionary of text attributes for the text at the provided character index.

Getting Attributes and Length

- [length](#) (page 9)
Returns the length of the text document.
- [validAttributesForMarkedText](#) (page 12)
Returns an array of names for the attributes supported by the receiver.

Managing Text Input

- [overrideKeyboardWithKeyboardNamed:](#) (page 10)
Overrides the current keyboard.
- [selectInputMode:](#) (page 10)
Selects a new input mode.
- [supportsUnicode](#) (page 11)
Tests to see if the current input session supports Unicode text.
- [bundleIdentifier](#) (page 7)
Returns the bundle identifier for the process that the input session is attached to.
- [windowLevel](#) (page 12)
Returns the window level for a client window.

Instance Methods

attributedStringFromRange:

Returns the attributed string for the provided range of text.

```
(NSAttributedString*)attributedStringFromRange:(NSRange)range
```

Parameters

range

The range of text, relative to the document, that specifies the string to retrieve.

Return Value

The attributed string. See the `CTStringAttributes.h` header file for the attributes that can be included in this string. If the client does not support the `TSMDocumentAccess` protocol, the returned string is created from data obtained by sending the client application a `kEventTextInputGetSelectedText` Carbon event. The returned `NSAttributedString` object is an autoreleased object.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [attributesForCharacterIndex:lineHeightRectangle:](#) (page 7)

Declared In

`IMKInputSession.h`

attributesForCharacterIndex:lineHeightRectangle:

Returns a dictionary of text attributes for the text at the provided character index.

```
-(NSDictionary*)attributesForCharacterIndex:(NSUInteger)index
lineHeightRectangle:(NSRect*)lineRect
```

Parameters

index

The character index whose attributes you want to retrieve. The index is relative to the inline session. Note that if there is no inline session the value of *index* should be 0, which indicates that the information should be taken from the current selection.

lineRect

On return, a rectangle that frames a one-pixel wide rectangle with the height of the line. This rectangle is oriented the same way the line is oriented.

Return Value

A dictionary that contains the text attributes for the text at the provided character index. The returned `NSDictionary` object is an autoreleased object. The attributes include the `CTFontRef` for the text at that index, and the text orientation. The text orientation is indicated by an `NSNumber` whose value is 0 if the text is vertically oriented and 1 if the text is horizontally oriented. The key for this value is `IMKTextOrientationKey`.

Discussion

An input method calls this method to place a candidate window on screen.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [characterIndexForPoint:tracking:inMarkedRange:](#) (page 8)

Declared In

`IMKInputSession.h`

bundleIdentifier

Returns the bundle identifier for the process that the input session is attached to.

```
-(NSString*)bundleIdentifier
```

Return Value

The bundle identifier for the process that the input session is attached to. The returned `NSString` is an autoreleased object.

Discussion

Many input methods need to be able to identify the process that input sessions belong to. This method provides that service.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`IMKInputSession.h`

characterIndexForPoint:tracking:inMarkedRange:

Returns the location in the text document that maps to a global point.

```
- (NSInteger)characterIndexForPoint:(NSPoint)point
   tracking:(IMKLocationToOffsetMappingMode)mappingMode
   inMarkedRange:(BOOL*)inMarkedRange
```

Parameters

point

The point to map. This is a global point, typically from a mouse down operation.

mappingMode

The mapping mode. If the input method is tracking the mouse, the mode should be `kIMKMouseTrackingMode`. If the input method simply wants to map a screen position to an offset, than set the mode to `kIMKNearestBoundaryMode`.

inMarkedRange

On return, if the point is inside the text body and inside the marked range, this parameter is set to YES. Otherwise, if the point is outside the marked range, then on return this parameter is set to NO.

Return Value

The location in the text document.

Discussion

If the input method is tracking the mouse, the application should highlight to the active inline area.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [attributesForCharacterIndex:lineHeightRectangle:](#) (page 7)

Declared In

IMKInputSession.h

insertText:replacementRange:

Sends fully converted text to an input session.

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

Parameters

string

The converted text.

replacementRange

The replacement range. This parameter allows input methods to insert text at a location other than the current selection. If you use it, the replacement range should be relative to the beginning of the client document. If the string should be inserted at the current selection specify a replacement range with a location and length of `NSNotFound`. If the client does not support the `TSMDocumentAccess` protocol, this method ignores the replacement range string.

Discussion

When an input method finishes a conversion it calls this method and passes the finished text as an `NSString` or `NSAttributedString` object.

Availability

Available in Mac OS X v10.5 and later.

Declared In

IMKInputSession.h

length

Returns the length of the text document.

- (NSInteger)length

Return Value

The text document length. If the client does not support the `TSMDocumentAccess` protocol the returned value is `NSNotFound`.

Discussion

This method can be computationally expensive depending on how the client stores text. For that reason, you should avoid calling this method frequently.

Availability

Available in Mac OS X v10.5 and later.

Declared In

IMKInputSession.h

markedRange

Returns the range in the document that is occupied by marked text, that is, the current inline session.

- (NSRange)markedRange

Return Value

The range of the marked text.

Discussion

While an input method is converting text input and sending it to the client, the client maintains a range of text that is marked. The marked text is underlined in certain ways to indicate to the user that their keystrokes are in the process of being converted by an input method and the conversion has not yet been finalized. The input method uses this method to request that range.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setMarkedText:selectionRange:replacementRange:](#) (page 11)

Declared In

IMKInputSession.h

overrideKeyboardWithKeyboardNamed:

Overrides the current keyboard.

```
-(void)overrideKeyboardWithKeyboardNamed:(NSString*)keyboardUniqueName
```

Parameters

keyboardUniqueName

A unique keyboard name.

Discussion

The client tries to locate a keyboard layout with that name in the input method bundle. If a layout is found it is passed to the client who then tells the text service manager to use that layout for keyboard events. Input methods should call the method each time they are activated. If an input method uses a system keyboard to override the current keyboard, they are responsible for determining the unique name of the keyboard. Typically this name a DNS type name such as: `com.apple.<some name>`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

IMKInputSession.h

selectedRange

Returns the current selection range.

```
-(NSRange)selectedRange
```

Return Value

The current selection range, relative to the client document. If the client does not support the `TSMDocumentAccess` protocol the returned range has a location value of `NSNotFound` and a length of `NSNotFound`.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setMarkedText:selectionRange:replacementRange:](#) (page 11)

Declared In

IMKInputSession.h

selectInputMode:

Selects a new input mode.

```
-(void)selectInputMode:(NSString*)modeIdentifier
```

Parameters

modeIdentifier

An `NSString` object with a DNS format, such as: `com.<company name>.inputmethod.<some name>`. The identifier should match one of the keys in the component input mode dictionary.

Discussion

This method allows an input method to change its mode directly.

Availability

Available in Mac OS X v10.5 and later.

Declared In

IMKInputSession.h

setMarkedText:selectionRange:replacementRange:

Inserts the provided text and marks it to indicate that it is part of an active input session.

```
- (void)setMarkedText:(id)string selectionRange:(NSRange)selectionRange
replacementRange:(NSRange)replacementRange
```

Parameters

string

An `NSString` or an `NSAttributedString` object. Passing an `NSString` object produces default marking. On a 72 dpi screen the default marking is a 2 pixel black underline for the entire string.

selectionRange

The selection range, which is relative to the `string` parameter. For example, if the string contains the these characters: "INPUT" and the range is (5,0), the selection is set immediately after the "T."

replacementRange

A range that specifies the location in the client document where the marked text should be placed. If the marked text should be placed at the current cursor location `replacementRange` equals `NSNotFound`. If not `NSNotFound`, the replacement range is relative to the client document (0 is the beginning of the document) and NOT the string parameter. Currently, the replacement range is sent to the client via the `kEventParamTextInputSendReplaceRange` Carbon Event parameter. For this parameter to be used the client must support the `TSMDocumentAccess` protocol. If a client does not support the `TSMDocumentAccess` protocol the `replacementRange` parameter is ignored.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [markedRange](#) (page 9)
- [selectedRange](#) (page 10)

Declared In

IMKInputSession.h

supportsUnicode

Tests to see if the current input session supports Unicode text.

```
-(BOOL)supportsUnicode
```

Return Value

YES if the current input session supports Unicode text; otherwise NO.

Discussion

Input methods that restrict the character codes sent to a client if that client does not support Unicode text should call this method to learn whether a given input session supports Unicode text.

Availability

Available in Mac OS X v10.5 and later.

Declared In

IMKInputSession.h

validAttributesForMarkedText

Returns an array of names for the attributes supported by the receiver.

- (NSArray*)validAttributesForMarkedText

Return Value

An array of attribute names. The returned `NSArray` should not be released unless it is first retained.

Discussion

Input methods should restrict the attributes used to create attributed strings to the attributes in this array.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [attributesForCharacterIndex:lineHeightRectangle:](#) (page 7)
- [markedRange](#) (page 9)

Declared In

IMKInputSession.h

windowLevel

Returns the window level for a client window.

- (CGWindowLevel>windowLevel

Return Value

The window level for a client window.

Discussion

Call this to determine the client window level. Internally, the `IMKCandidates` class uses this method to determine the correct level for candidate display. Therefore, input methods that use `IMKCandidates` have no reason to call this method. However, many input method developers build their own candidate display system, and this method is provided for their use. In order to display a candidate window at the correct level, use this method to obtain the client window level, increment the returned level, and then apply that level to any candidate windows.

Availability

Available in Mac OS X v10.5 and later.

Declared In

IMKInputSession.h

Constants

IMKLocationToOffsetMappingMode

List modes for mapping a screen location to a text offset.

```
enum { kIMKMouseTrackingMode = (1<<0), kIMKNearestBoundaryMode = (1<<1) };typedef
NSInteger IMKLocationToOffsetMappingMode;
```

Constants

kIMKMouseTrackingMode

Mouse tracking mode. When tracking, a coordinate does not change the offset until the coordinate is completely past a given character.

Available in Mac OS X v10.5 and later.

Declared in IMKInputSession.h.

kIMKNearestBoundaryMode

Nearest boundary mode. When trying to place an insertion point, a coordinate is mapped to the nearest character boundary.

Available in Mac OS X v10.5 and later.

Declared in IMKInputSession.h.

Declared In

IMKInputSession.h

IMKTextOrientationName

The text orientation name key.

```
extern const NSString* IMKTextOrientationName;
```

Constants

IMKTextOrientationName

The key used to find the client text orientation value, which is part of the dictionary returned by the method [attributesForCharacterIndex:lineHeightRectangle:](#) (page 7). The associated value is an `NSNumber` object that represents a `BOOL` value. A value of 1 means the client text uses a horizontal layout. A value of 0 means that the client text uses a vertical layout.

Available in Mac OS X v10.5 and later.

Declared in IMKInputSession.h.

Declared In

IMKInputSession.h

Document Revision History

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

Date	Notes
2007-06-05	New document that describes the protocol used to send and obtain data from an input method client session.

REVISION HISTORY

Document Revision History

Index

A

attributedStringFromRange: protocol instance method [6](#)

attributesForCharacterIndex:lineHeightRectangle: protocol instance method [7](#)

B

bundleIdentifier protocol instance method [7](#)

C

characterIndexForPoint:tracking:inMarkedRange: protocol instance method [8](#)

I

IMKLocationToOffsetMappingMode [13](#)

IMKTextOrientationName [13](#)

IMKTextOrientationName constant [13](#)

insertText:replacementRange: protocol instance method [8](#)

K

kIMKMouseTrackingMode constant [13](#)

kIMKNearestBoundaryMode constant [13](#)

L

length protocol instance method [9](#)

M

markedRange protocol instance method [9](#)

O

overrideKeyboardWithKeyboardNamed: protocol instance method [10](#)

S

selectedRange protocol instance method [10](#)

selectInputMode: protocol instance method [10](#)

setMarkedText:selectionRange:replacementRange: protocol instance method [11](#)

supportsUnicode protocol instance method [11](#)

V

validAttributesForMarkedText protocol instance method [12](#)

W

windowLevel protocol instance method [12](#)