
NSInputManager Class Reference

[Cocoa](#) > [Events & Other Input](#)



2007-04-01



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, 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

NSInputManager Class Reference 5

Overview	5
Adopted Protocols	5
Tasks	6
Creating Input Managers	6
Getting Active Input Manager	6
Handling Events	6
Getting Input Server Information	6
Deprecated	7
Class Methods	7
currentInputManager	7
cycleToNextInputLanguage:	7
cycleToNextInputServerInLanguage:	7
Instance Methods	8
handleMouseEvent:	8
image	8
initWithName:host:	8
language	9
localizedInputManagerName	9
markedTextAbandoned:	9
markedTextSelectionChanged:client:	10
server	10
wantsToDelayTextChangeNotifications	10
wantsToHandleMouseEvents	11
wantsToInterpretAllKeystrokes	11

Document Revision History 13

Index 15

NSInputManager Class Reference

Inherits from	NSObject
Conforms to	NSTextInput NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Text Input Management
Declared in	NSInputManager.h

Overview

NSInputManager is one of the three players in the Cocoa text input management system. The input manager acts as a proxy between "NSInputServer" and text views. You never have to instantiate or subclass NSInputManager, and unless you are implementing a text view that does not inherit from NSTextView, you never have to directly access its methods either.

If an application needs more complex text handling than the standard Cocoa text view classes can provide, then it will use its own text view class that implements the NSTextInput protocol, and that class will call NSInputManager's methods. The current input manager's input server will call some of the text view's NSTextInput protocol methods.

The [wantsToDelayTextChangeNotifications](#) (page 10), [wantsToHandleMouseEvents](#) (page 11), and [wantsToInterpretAllKeystrokes](#) (page 11) methods call methods of the same names on the input manager's current input server and return the result.

Adopted Protocols

NSTextInput

- characterIndexForPoint:
- conversationIdentifier
- doCommandBySelector:
- firstRectForCharacterRange:
- hasMarkedText
- insertText:
- markedRange

- `selectedRange`
- `setMarkedText:selectedRange:`
- `unmarkText`
- `validAttributesForMarkedText`

Tasks

Creating Input Managers

- `initWithName:host:` (page 8)

Getting Active Input Manager

- + `currentInputManager` (page 7)
The current input manager is the one that has been chosen to handle keyboard events at the time this method is called.

Handling Events

- `markedTextAbandoned:` (page 9)
The input server must abandon whatever it was doing with marked text.
- `markedTextSelectionChanged:client:` (page 10)
The user clicked at the beginning of marked text, the end of marked text, or in between, or the user made a selection within the marked text in the *client* text view.
- `wantsToDelayTextChangeNotifications` (page 10)
A YES return value tells the sender that only a call to its `insertText:client:` method constitutes a modification to its text storage.
- `wantsToInterpretAllKeystrokes` (page 11)
Returns YES if the server wants all keystrokes to be sent to it as characters.
- `wantsToHandleMouseEvents` (page 11)
Returns YES if the sender should forward all mouse events within the text view to the input server.
- `handleMouseEvent:` (page 8)

Getting Input Server Information

- `image` (page 8)
Deprecated.
- `language` (page 9)
Returns the `Language` property from the input server's `Info` file, or `nil` if none is specified there.

- [localizedInputManagerName](#) (page 9)
Returns the name of the input server as it appears in the Edit>Input submenu.
- [server](#) (page 10)
Deprecated.

Deprecated

- + [cycleToNextInputLanguage:](#) (page 7)
Deprecated.
- + [cycleToNextInputServerInLanguage:](#) (page 7)
Deprecated.

Class Methods

currentInputManager

The current input manager is the one that has been chosen to handle keyboard events at the time this method is called.

```
+ (NSInputManager *)currentInputManager
```

Discussion

Don't cache the return value, because the user can switch to a different input manager at any time.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSInputManager.h

cycleToNextInputLanguage:

Deprecated.

```
+ (void)cycleToNextInputLanguage:(id)sender
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSInputManager.h

cycleToNextInputServerInLanguage:

Deprecated.

```
+ (void)cycleToNextInputServerInLanguage:(id)sender
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSInputManager.h

Instance Methods

handleMouseEvent:

```
- (BOOL)handleMouseEvent:(NSEvent *)theMouseEvent
```

Discussion

Forwards a mouse event passed in *theMouseEvent* to the input server. If [wantsToHandleMouseEvents](#) (page 11) returns YES, then the text view must forward all mouse events that occur within it. As usual, a return value of NO means that the text view should proceed with handling the event.

For additional information see the NSInputServerMouseTracker class.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSInputManager.h

image

Deprecated.

```
- (NSImage *)image
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSInputManager.h

initWithName:host:

```
- (NSInputManager *)initWithName:(NSString *)inputServerName host:(NSString *)hostName
```

Discussion

Cocoa constructs an NSInputManager automatically. This method is an implementation detail. You will never call this method.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSInputManager.h

language

Returns the `Language` property from the input server's `Info` file, or `nil` if none is specified there.

- (NSString *)language

Discussion

For additional information, see “Deploying Input Servers”.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSInputManager.h

localizedInputManagerName

Returns the name of the input server as it appears in the Edit>Input submenu.

- (NSString *)localizedInputManagerName

Discussion

This value comes from the input server's `Info` file. For additional information, see “Deploying Input Servers”.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSInputManager.h

markedTextAbandoned:

The input server must abandon whatever it was doing with marked text.

- (void)markedTextAbandoned:(id)client

Discussion

The `NSTextView` object `client` calls this when the user clicks outside the marked text (anywhere other than the beginning of marked text, the end of marked text, or in between), then `NSTextView` promotes the marked text to normal text as if it had been inserted. A custom text view is free to choose not to keep the marked text.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [markedTextSelectionChanged:client:](#) (page 10)
- `markedTextAbandoned:` (`NSInputServiceProvider`)

Declared In

NSInputManager.h

markedTextSelectionChanged:client:

The user clicked at the beginning of marked text, the end of marked text, or in between, or the user made a selection within the marked text in the *client* text view.

```
- (void)markedTextSelectionChanged:(NSRange)newSel client:(id)client
```

Discussion

The range *newSel* is relative to the beginning of the marked text.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [markedTextAbandoned:](#) (page 9)
- [markedTextSelectionChanged:client:](#) (NSInputServiceProvider)

Declared In

NSInputManager.h

server

Deprecated.

```
- (NSInputServer *)server
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSInputManager.h

wantsToDelayTextChangeNotifications

A YES return value tells the sender that only a call to its `insertText:client:` method constitutes a modification to its text storage.

```
- (BOOL)wantsToDelayTextChangeNotifications
```

Discussion

A NO return value tells the sender that all text given to it, either by insertion or as part of marked text, should constitute a modification to its text storage. The sender may for example want to filter all text that is part of a modification but leave marked text unfiltered.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [wantsToDelayTextChangeNotifications](#) (page 10) (NSInputServiceProvider)

Declared In

NSInputManager.h

wantsToHandleMouseEvents

Returns YES if the sender should forward all mouse events within the text view to the input server.

- (BOOL)wantsToHandleMouseEvents

Availability

Available in Mac OS X v10.0 and later.

See Also

- wantsToHandleMouseEvents (NSInputServiceProvider)

Declared In

NSInputManager.h

wantsToInterpretAllKeystrokes

Returns YES if the server wants all keystrokes to be sent to it as characters.

- (BOOL)wantsToInterpretAllKeystrokes

Discussion

This method is needed only by the inner workings of Cocoa. You will probably not need to call this method.

Availability

Available in Mac OS X v10.0 and later.

See Also

- wantsToInterpretAllKeystrokes (NSInputServiceProvider)

Declared In

NSInputManager.h

Document Revision History

This table describes the changes to *NSInputManager Class Reference*.

Date	Notes
2007-04-01	Made editorial improvements.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

C

currentInputManager **class method** [7](#)
cycleToNextInputLanguage: **class method** [7](#)
cycleToNextInputServerInLanguage: **class method**
[7](#)

H

handleMouseEvent: **instance method** [8](#)

I

image **instance method** [8](#)
initWithName:host: **instance method** [8](#)

L

language **instance method** [9](#)
localizedInputManagerName **instance method** [9](#)

M

markedTextAbandoned: **instance method** [9](#)
markedTextSelectionChanged:client: **instance**
method [10](#)

S

server **instance method** [10](#)

W

wantsToDelayTextChangeNotifications **instance**
method [10](#)
wantsToHandleMouseEvents **instance method** [11](#)
wantsToInterpretAllKeystrokes **instance method**
[11](#)