
IMKCandidates Class 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, 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

IMKCandidates Class Reference 5

Overview	5
Tasks	5
Initializing a Candidates Window	5
Managing Selection Keys	5
Managing Window Visibility and Behavior	6
Managing Window Type and Text Attributes	6
Showing an Annotation Window	6
Instance Methods	7
attributes	7
dismissesAutomatically	7
hide	7
initWithServer:panelType:	8
isVisible	8
panelType	8
selectionKeys	9
selectionKeysKeylayout	9
setAttributes:	10
setDismissesAutomatically:	10
setPanelType:	11
setSelectionKeys:	11
setSelectionKeysKeylayout:	12
show:	12
showAnnotation:	13
updateCandidates	13
Constants	14
IMKCandidatePanelType	14
IMKCandidatesLocationHint	14
IMKCandidatesOpacityAttributeName	15

Document Revision History 17

Index 19

IMKCandidates Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	System/Library/Frameworks/InputMethodKit.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	IMKCandidates.h
Related sample code	NumberInput_IMKit_Sample

Overview

The `IMKCandidates` class presents candidates to users and notifies the appropriate `IMKInputController` object when the user selects a candidate. **Candidates** are alternate characters for a given input sequence. The `IMKCandidates` class supports using a candidates window in your input method; using `IMKCandidates` is optional. Not all input methods require them.

When you create an `IMKCandidates` object, you attach it to the `IMKServer` object for your input method. You then need to override the `IMKInputController` methods `candidateSelectionChanged:` and `candidateSelected:` as well as implement a candidates method in your delegate object. The `IMKInputController` subclass supplies candidates to the `IMKCandidates` object by implementing the candidates method. When you are ready to display a candidates window, call the candidates method to update candidates and to show the candidates window.

Tasks

Initializing a Candidates Window

- [initWithServer:panelType:](#) (page 8)
Returns the initialized `IMKCandidates` object.

Managing Selection Keys

- [setSelectionKeys:](#) (page 11)
Sets the selection keys for the candidates.

- [selectionKeys](#) (page 9)
Returns an array of `NSNumber` objects where each `NSNumber` object represents a virtual key code.
- [setSelectionKeysKeyLayout:](#) (page 12)
Sets the key layout that is used to map virtual key codes to characters.
- [selectionKeysKeyLayout](#) (page 9)
Returns the key layout that maps virtual key codes to selection keys.

Managing Window Visibility and Behavior

- [show:](#) (page 12)
Shows the candidates window.
- [hide](#) (page 7)
Hides a candidates window, if it is visible.
- [isVisible](#) (page 8)
Returns whether or not the candidates window is visible.
- [setDismissesAutomatically:](#) (page 10)
Sets the state of the flag that determines whether the candidates window dismisses automatically.
- [dismissesAutomatically](#) (page 7)
Returns the state of the flag that determines whether the candidates window dismisses automatically.
- [updateCandidates](#) (page 13)
Updates the candidates that are displayed in the candidates window.

Managing Window Type and Text Attributes

- [panelType](#) (page 8)
Returns the style of the candidates window.
- [setPanelType:](#) (page 11)
Sets the style of the candidates window.
- [setAttributes:](#) (page 10)
Sets the style attributes for the candidates window.
- [attributes](#) (page 7)
Returns a dictionary of the style attributes used for the candidates window..

Showing an Annotation Window

- [showAnnotation:](#) (page 13)
Displays an annotation string in an annotation window.

Instance Methods

attributes

Returns a dictionary of the style attributes used for the candidates window..

- (NSDictionary *)attributes

Return Value

The dictionary that contains the keys and values for the styles.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setAttribute:](#) (page 10)

Declared In

IMKCandidates.h

dismissesAutomatically

Returns the state of the flag that determines whether the candidates window dismisses automatically.

- (BOOL)dismissesAutomatically

Return Value

YES if the candidates window dismisses automatically; otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setDismissesAutomatically:](#) (page 10)

Declared In

IMKCandidates.h

hide

Hides a candidates window, if it is visible.

- (void)hide

Availability

Available in Mac OS X v10.5 and later.

See Also

- [show:](#) (page 12)

- [isVisible:](#) (page 8)

Declared In

IMKCandidates.h

initWithServer:panelType:

Returns the initialized IMKCandidates object.

- (id)initWithServer:(IMKServer *)server panelType:(IMKCandidatePanelType)panelType

Parameters*server*

The IMKServer object that manages the candidate and the panel type.

panelType

A panel type for the candidate window.

Return Value

The initialized IMKCandidates object.

Discussion

When an input method allocates an IMKCandidates object it should initialize that object by calling this method.

Availability

Available in Mac OS X v10.5 and later.

Declared In

IMKCandidates.h

isVisible

Returns whether or not the candidates window is visible.

- (BOOL)isVisible

Return Value

YES if the candidates window is visible; otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [hide](#) (page 7)
- [show:](#) (page 12)

Declared In

IMKCandidates.h

panelType

Returns the style of the candidates window.

- (IMKCandidatePanelType)panelType

Return Value

A “[IMKCandidatePanelType](#)” (page 14) constant that represents the style of the candidates window.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setPanelType:](#) (page 11)

Declared In

IMKCandidates.h

selectionKeys

Returns an array of `NSNumber` objects where each `NSNumber` object represents a virtual key code.

- (NSArray *)selectionKeys

Return Value

The array of `NSNumber` objects. This array is an autorelease object. Do not release it unless you first retain it.

Discussion

Selection keys are keys that can be used to select one of the candidates. They are displayed next to the candidate that will be selected when the user types that key.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setSelectionKeys:](#) (page 11)

Declared In

IMKCandidates.h

selectionKeysKeylayout

Returns the key layout that maps virtual key codes to selection keys.

- (TISInputSourceRef)selectionKeysKeylayout

Return Value

The key layout in use. By default this is the key layout whose source id is `com.apple.keylayout.US`. This object is an autorelease object. Do not release it unless you first retain it.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setSelectionKeysKeylayout:](#) (page 12)

Declared In

IMKCandidates.h

setAttributes:

Sets the style attributes for the candidates window.

```
- (void)setAttributes:(NSDictionary *)attributes
```

Parameters

attributes

A dictionary that contains keys and values for the styles to use. You can supply the keys and values listed in the following table:

Key	Value
NSFontAttributeName	An <code>NSFont</code> object. Setting the font attribute sets the font that is used to draw Candidates. It does not effect the selection keys which are always drawn in the same font. Note that to set the font size you should use this key/value pair.
IMKCandidatesOpacityAttributeName (page 15)	An <code>NSNumber</code> object that represents a floating-point value between 0.0 (transparent) and 1.0 (completely opaque). The default opacity is 1.0.
NSForegroundColorAttributeName	An <code>NSColor</code> object to use for the candidate text color. The default color is black.
NSBackgroundColorDocumentAttribute	An <code>NSColor</code> object to use for the background color behind the candidate text.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [attributes](#) (page 7)

Declared In

IMKCandidates.h

setDismissesAutomatically:

Sets the state of the flag that determines whether the candidates window dismisses automatically.

```
- (void)setDismissesAutomatically:(BOOL)flag
```

Parameters

flag

YES to have the candidates window dismiss automatically; otherwise NO.

Discussion

By default, if the user presses the Return or Enter keys, the candidates are dismissed and a `candidateSelected:` message is sent to the input controller. You can call the `setDismissesAutomatically:` method, passing NO as the `flag` parameter to change the default dismissal behavior. The input controller still receives a `candidateSelected:` message.

When you set the flag to `NO`, an input method processes text input while dynamically updating the content of the candidates as the user inputs text. When a session deactivates, candidate window is hidden regardless of the state of the flag.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [dismissesAutomatically](#) (page 7)

Declared In

IMKCandidates.h

setPanelType:

Sets the style of the candidates window.

```
- (void)setPanelType:(IMKCandidatePanelType)panelType
```

Parameters

panelType

A “[IMKCandidatePanelType](#)” (page 14) constant that represents the style of the candidates window.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [panelType](#) (page 8)

Related Sample Code

NumberInput_IMKit_Sample

Declared In

IMKCandidates.h

setSelectionKeys:

Sets the selection keys for the candidates.

```
- (void)setSelectionKeys:(NSArray *)keyCodes
```

Parameters

keyCodes

An array of `NSNumber` objects where each `NSNumber` object represents a virtual key code. The input controller maps these key codes to characters that are displayed either across the top of the candidates, if the candidates are laid out horizontally, or along the left edge of the candidates, if they are aligned vertically.

Discussion

Selection keys are keys that can be used to select one of the candidates. They are displayed next to the candidate that will be selected when the user types that key.

The number of selection keys determines how many candidates are displayed per page. For example, if you pass an array of four key codes, four candidates are displayed per page. If you pass eleven key codes, eleven candidates are displayed. By default, the key codes are mapped using the keyboard layout whose source id is `com.apple.keyboard.US`. You can replace the default layout by calling [setSelectionKeysKeylayout:](#) (page 12). The default selection keys are the digits 1 through 9 or, in terms of key codes, 18, 19, 20, 21, 23, 22, 26, 28, and 25.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [selectionKeys](#) (page 9)

Declared In

IMKCandidates.h

setSelectionKeysKeylayout:

Sets the key layout that is used to map virtual key codes to characters.

```
- (void)setSelectionKeysKeylayout:(TISInputSourceRef) layout
```

Parameters

layout

The key layout to use.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [selectionKeysKeylayout](#) (page 9)

Declared In

IMKCandidates.h

show:

Shows the candidates window.

```
- (void)show:(IMKCandidatesLocationHint) locationHint
```

Parameters

locationHint

A “[IMKCandidatesLocationHint](#)” (page 14) constant that specifies the desired position of the candidates window. The Input Method Kit uses the hint to place the candidates window in a location that is in the vicinity of the hint location and ensures that the candidates window is fully visible.

Discussion

Your input method calls this method when it is appropriate during text conversion to display a list of candidates.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [hide](#) (page 7)
- [isVisible](#) (page 8)

Related Sample Code

NumberInput_IMKit_Sample

Declared In

IMKCandidates.h

showAnnotation:

Displays an annotation string in an annotation window.

```
- (void)showAnnotation:(NSAttributedString *)annotationString
```

Parameters

annotationString

The string to display.

Discussion

An annotation string explains or comments on the candidate string in the candidates window. An annotation window is a small, borderless window that is aligned with the current candidates window. An input method calls `showAnnotation:` when the `candidateSelectionChanged:` method of the `IMKInputController` class is called, and the candidate string has annotations.

Availability

Available in Mac OS X v10.5 and later.

Declared In

IMKCandidates.h

updateCandidates

Updates the candidates that are displayed in the candidates window.

```
- (void)updateCandidates
```

Discussion

When you call this method, the Input Method Kit calls the `candidates` method of the `IMKInputController` class. Note that the candidates list is updated, but the visible state of the window does not change. In other words, if the window is hidden, it remains hidden. If the window is visible, it remains visible.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

NumberInput_IMKit_Sample

Declared In

IMKCandidates.h

Constants

IMKCandidatePanelType

Types of candidates windows provide by the Input Method Kit.

```
enum {    kIMKSingleColumnScrollingCandidatePanel = 1,
          kIMKScrollingGridCandidatePanel = 2,
          kIMKSingleRowSteppingCandidatePanel = 3 };
typedef NSUInteger IMKCandidatePanelType;
```

Constants

`kIMKSingleColumnScrollingCandidatePanel`
 A window that displays one column and can scroll if necessary.
 Available in Mac OS X v10.5 and later.
 Declared in `IMKCandidates.h`.

`kIMKScrollingGridCandidatePanel`
 A window that displays a grid and can scroll if necessary.
 Available in Mac OS X v10.5 and later.
 Declared in `IMKCandidates.h`.

`kIMKSingleRowSteppingCandidatePanel`
 A window that displays a single row.
 Available in Mac OS X v10.5 and later.
 Declared in `IMKCandidates.h`.

Declared In

`IMKCandidates.h`

IMKCandidatesLocationHint

Hints that suggest where to place the candidates window.

```
enum {    kIMKLocateCandidatesAboveHint = 1,
          kIMKLocateCandidatesBelowHint = 2,
          kIMKLocateCandidatesLeftHint = 3,
          kIMKLocateCandidatesRightHint = 4
};typedef NSUInteger IMKCandidatesLocationHint;
```

Constants

`kIMKLocateCandidatesAboveHint`
 Place the candidates window above the start of the current text selection.
 Available in Mac OS X v10.5 and later.
 Declared in `IMKCandidates.h`.

`kIMKLocateCandidatesBelowHint`
 Place the candidates window below the start of the current text selection.
 Available in Mac OS X v10.5 and later.
 Declared in `IMKCandidates.h`.

`kIMKLocateCandidatesLeftHint`

Place the candidates window to the left of the current text selection.

Available in Mac OS X v10.5 and later.

Declared in `IMKCandidates.h`.

`kIMKLocateCandidatesRightHint`

Place the candidates window to the right of the current text selection.

Available in Mac OS X v10.5 and later.

Declared in `IMKCandidates.h`.

Discussion

The Input Method Kit uses the hint to place the candidates window in a location that is in the vicinity of the hint location, but that also ensures that the candidates window is fully visible.

Declared In

`IMKCandidates.h`

IMKCandidatesOpacityAttributeName

The opacity level for a candidates window.

```
extern const NSString* IMKCandidatesOpacityAttributeName;
```

Constants

`IMKCandidatesOpacityAttributeName`

The opacity attribute for a candidates window. The associated value must be an `NSNumber` object that represents a value from 0 to 1.

Available in Mac OS X v10.5 and later.

Declared in `IMKCandidates.h`.

Declared In

`IMKCandidates.h`

Document Revision History

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

Date	Notes
2007-06-05	New document that describes the class that supports the use of candidate windows for an input method.

REVISION HISTORY

Document Revision History

Index

A

attributes [instance method 7](#)

D

dismissesAutomatically [instance method 7](#)

H

hide [instance method 7](#)

I

IMKCandidatePanelType [14](#)

IMKCandidatesLocationHint [14](#)

IMKCandidatesOpacityAttributeName [15](#)

IMKCandidatesOpacityAttributeName [constant 15](#)

initWithServer:panelType: [instance method 8](#)

isVisible [instance method 8](#)

K

kIMKLocateCandidatesAboveHint [constant 14](#)

kIMKLocateCandidatesBelowHint [constant 14](#)

kIMKLocateCandidatesLeftHint [constant 15](#)

kIMKLocateCandidatesRightHint [constant 15](#)

kIMKScrollingGridCandidatePanel [constant 14](#)

kIMKSingleColumnScrollingCandidatePanel
[constant 14](#)

kIMKSingleRowSteppingCandidatePanel [constant
14](#)

P

panelType [instance method 8](#)

S

selectionKeys [instance method 9](#)

selectionKeysKeylayout [instance method 9](#)

setAttributes: [instance method 10](#)

setDismissesAutomatically: [instance method 10](#)

setPanelType: [instance method 11](#)

setSelectionKeys: [instance method 11](#)

setSelectionKeysKeylayout: [instance method 12](#)

showAnnotation: [instance method 13](#)

show: [instance method 12](#)

U

updateCandidates [instance method 13](#)