
UIWindow Class Reference

User Experience: Windows & Views



2010-05-06



Apple Inc.
© 2010 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, iPhone, and Objective-C are trademarks of Apple Inc., registered in the United States and other countries.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

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

UIWindow Class Reference 5

Overview	5
Tasks	5
Configuring Windows	5
Making Windows Key	6
Converting Coordinates	6
Sending Events	6
Properties	6
keyWindow	6
rootViewController	7
screen	7
windowLevel	8
Instance Methods	8
becomeKeyWindow	8
convertPoint:fromWindow:	8
convertPoint:toWindow:	9
convertRect:fromWindow:	9
convertRect:toWindow:	10
makeKeyAndVisible	10
makeKeyWindow	11
resignKeyWindow	11
sendEvent:	12
Constants	12
UIWindowLevel	12
Keyboard Notification User Info Keys	12
Notifications	14
UIWindowDidBecomeVisibleNotification	14
UIWindowDidBecomeHiddenNotification	14
UIWindowDidBecomeKeyNotification	15
UIWindowDidResignKeyNotification	15
UIKeyboardWillShowNotification	15
UIKeyboardDidShowNotification	15
UIKeyboardWillHideNotification	16
UIKeyboardDidHideNotification	16

Document Revision History 17

UIWindow Class Reference

Inherits from	UIView : UIResponder : NSObject
Conforms to	NSCoding (UIView) NSObject (NSObject)
Framework	/System/Library/Frameworks/UIKit.framework
Availability	Available in iOS 2.0 and later.
Declared in	UIWindow.h
Related sample code	GKRocket GKTank MoviePlayer ScrollViewSuite SpeakHere

Overview

The `UIWindow` class defines objects (known as **windows**) that manage and coordinate the windows an application displays on the screen. The two principal functions of a window are to provide an area for displaying its views and to distribute events to the views. The window is the root view in the view hierarchy. A window belongs to a level; the windows in one level appear above another level. For example, alerts appear above normal windows. Typically, there is only one window in an iOS application.

Read *Windows and Views* in *iOS Application Programming Guide* to learn how to use this class.

Tasks

Configuring Windows

[windowLevel](#) (page 8) *property*

The receiver's window level.

[screen](#) (page 7) *property*

The screen on which the window is currently displayed.

[rootViewController](#) (page 7) *property*

The root view controller for the window.

Making Windows Key

`keyWindow` (page 6) *property*

A Boolean value that indicates whether the receiver is the key window for the application. (read-only)

- `makeKeyAndVisible` (page 10)

Makes the receiver the key window and makes that window visible.

- `becomeKeyWindow` (page 8)

Invoked automatically to inform the receiver that it has become the key window; never invoke this method directly.

- `makeKeyWindow` (page 11)

Makes the receiver the main window.

- `resignKeyWindow` (page 11)

Invoked automatically when the window resigns key window status; never invoke this method directly.

Converting Coordinates

- `convertPoint:toWindow:` (page 9)

Converts a point from the receiver's coordinate system to that of another window.

- `convertPoint:fromWindow:` (page 8)

Converts a point from the coordinate system of a given window to that of the receiver.

- `convertRect:toWindow:` (page 10)

Converts a rectangle from the receiver's coordinate system to that of another window.

- `convertRect:fromWindow:` (page 9)

Converts a rectangle from the coordinate system of another window to that of the receiver.

Sending Events

- `sendEvent:` (page 12)

Dispatches events sent to the receiver by the `UIApplication` object to its views.

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

keyWindow

A Boolean value that indicates whether the receiver is the key window for the application. (read-only)

```
@property(nonatomic, readonly, getter=isKeyWindow) BOOL keyWindow
```

Discussion

If YES, the receiver is the key window for the application; otherwise, NO.

Availability

Available in iOS 2.0 and later.

See Also

- [makeKeyAndVisible](#) (page 10)
- [becomeKeyWindow](#) (page 8)
- [makeKeyWindow](#) (page 11)
- [resignKeyWindow](#) (page 11)

Declared In

UIWindow.h

rootViewController

The root view controller for the window.

```
@property(n nonatomic, retain) UIViewController *rootViewController
```

Discussion

The root view controller provides the content view of the window. Assigning a view controller to this property (either programmatically or using Interface Builder) installs the view controller's view as the content view of the window. If the window has an existing view hierarchy, the old views are removed before the new ones are installed.

The default value of this property is `nil`.

Availability

Available in iOS 4.0 and later.

Declared In

UIWindow.h

screen

The screen on which the window is currently displayed.

```
@property (nonatomic, retain) UIScreen *screen
```

Discussion

By default, all windows are created on the main screen. If additional screens are attached to the device, assigning a different screen object to this property causes the window to be displayed on the new screen.

Moving windows from screen to screen is a relatively expensive operation and should not be done in performance-sensitive code. Instead, it is recommended that you change the screen before displaying the window the first time. Changing the screen of a window that has not yet been ordered onto the screen has no significant additional cost.

Availability

Available in iOS 3.2 and later.

Declared In

UIWindow.h

windowLevel

The receiver's window level.

```
@property(nonatomic) UIWindowLevel windowLevel
```

Discussion

Levels are ordered so that each level groups windows within it in front of those in all preceding groups. For example, alert windows appear in front of all normal-level windows. When a window enters a new level, it's ordered in front of all its peers in that level. See “[UIWindowLevel](#)” (page 12) for a list of possible values. The default value is 0.0.

Availability

Available in iOS 2.0 and later.

Declared In

UIWindow.h

Instance Methods

becomeKeyWindow

Invoked automatically to inform the receiver that it has become the key window; never invoke this method directly.

```
- (void)becomeKeyWindow
```

Discussion

This method reestablishes the receiver's first responder, sends the [becomeKeyWindow](#) (page 8) message to that object if it responds, and posts [UIWindowDidBecomeKeyNotification](#) (page 15) to the default notification center.

Availability

Available in iOS 2.0 and later.

See Also

- [@property keyWindow](#) (page 6)
- [makeKeyAndVisible](#) (page 10)
- [makeKeyWindow](#) (page 11)
- [resignKeyWindow](#) (page 11)

Declared In

UIWindow.h

convertPoint:fromWindow:

Converts a point from the coordinate system of a given window to that of the receiver.

```
- (CGPoint)convertPoint:(CGPoint)point fromWindow:(UIWindow *)window
```


Parameters*point*A point specifying a location in the coordinate system of *window*.*window*The window with *point* in its coordinate system. If *nil*, this method converts the point from the logical coordinate system of the screen, which is measured in points.**Return Value**

The point converted to the coordinate system of the receiver.

Availability

Available in iOS 2.0 and later.

See Also- [convertPoint:toWindow:](#) (page 9)**Declared In**

UIWindow.h

convertPoint:toWindow:

Converts a point from the receiver's coordinate system to that of another window.

- (CGPoint)convertPoint:(CGPoint)*point* toWindow:(UIWindow *)*window***Parameters***point*

A point specifying a location in the logical coordinate system of the receiver.

*window*The window into whose coordinate system *point* is to be converted. If *nil*, this method converts the point to the logical coordinate system of the screen, which is measured in points.**Return Value**The point converted to the coordinate system of *window*.**Availability**

Available in iOS 2.0 and later.

See Also- [convertPoint:fromWindow:](#) (page 8)**Declared In**

UIWindow.h

convertRect:fromWindow:

Converts a rectangle from the coordinate system of another window to that of the receiver.

- (CGRect)convertRect:(CGRect)*rect* fromWindow:(UIWindow *)*window*

Parameters*rect*

The rectangle in the window's coordinate system.

window

The window with *rect* in its coordinate system. If *nil*, this method instead converts the rectangle from the logical coordinate system of the screen, which is measured in points.

Return Value

The converted rectangle.

Availability

Available in iOS 2.0 and later.

See Also

- [convertRect:toWindow:](#) (page 10)

Declared In

UIWindow.h

convertRect:toWindow:

Converts a rectangle from the receiver's coordinate system to that of another window.

```
- (CGRect)convertRect:(CGRect)rect toWindow:(UIWindow *)window
```

Parameters*rect*

A rectangle in the receiver's coordinate system.

window

The window that is the target of the conversion operation. If *nil*, this method instead converts the rectangle to the logical coordinate system of the screen, which is measured in points.

Return Value

The converted rectangle.

Availability

Available in iOS 2.0 and later.

See Also

- [convertRect:fromWindow:](#) (page 9)

Declared In

UIWindow.h

makeKeyAndVisible

Makes the receiver the key window and makes that window visible.

```
- (void)makeKeyAndVisible
```

Discussion

This is a convenience method to make the receiver the main window and displays it in front of other windows. You can also hide and reveal a window using the inherited `hiddenUIView` property.

Availability

Available in iOS 2.0 and later.

See Also

- [@property keyWindow](#) (page 6)
- [becomeKeyWindow](#) (page 8)
- [makeKeyWindow](#) (page 11)
- [resignKeyWindow](#) (page 11)

Declared In

UIWindow.h

makeKeyWindow

Makes the receiver the main window.

- (void)makeKeyWindow

Availability

Available in iOS 2.0 and later.

See Also

- [@property keyWindow](#) (page 6)
- [makeKeyAndVisible](#) (page 10)
- [becomeKeyWindow](#) (page 8)
- [resignKeyWindow](#) (page 11)

Declared In

UIWindow.h

resignKeyWindow

Invoked automatically when the window resigns key window status; never invoke this method directly.

- (void)resignKeyWindow

Discussion

This method sends [resignKeyWindow](#) (page 11) to the receiver's first responder and posts [UIWindowDidResignKeyNotification](#) (page 15) to the default notification center.

Availability

Available in iOS 2.0 and later.

See Also

- [@property keyWindow](#) (page 6)
- [makeKeyAndVisible](#) (page 10)
- [becomeKeyWindow](#) (page 8)
- [makeKeyWindow](#) (page 11)

Declared In

UIWindow.h

sendEvent:

Dispatches events sent to the receiver by the `UIApplication` object to its views.

```
- (void)sendEvent:(UIEvent *)event
```

Parameters

event

The event to process.

Availability

Available in iOS 2.0 and later.

Declared In

UIWindow.h

Constants

UIWindowLevel

The positioning of windows relative to each other.

```
const UIWindowLevel UIWindowLevelNormal;
const UIWindowLevel UIWindowLevelAlert;
const UIWindowLevel UIWindowLevelStatusBar;
typedef CGFloat UIWindowLevel;
```

Constants

UIWindowLevelNormal

The default level.

Available in iOS 2.0 and later.

Declared in UIWindow.h.

UIWindowLevelAlert

The level for an alert view.

Available in iOS 2.0 and later.

Declared in UIWindow.h.

UIWindowLevelStatusBar

The level for a status window.

Available in iOS 2.0 and later.

Declared in UIWindow.h.

Discussion

The stacking of levels takes precedence over the stacking of windows within each level. That is, even the bottom window in a level obscures the top window of the next level down. Levels are listed in order from lowest to highest.

Keyboard Notification User Info Keys

Keys used to get values from the user information dictionary of keyboard notifications.

```

NSString *const UIKeyboardFrameBeginUserInfoKey;
NSString *const UIKeyboardFrameEndUserInfoKey;
NSString *const UIKeyboardAnimationDurationUserInfoKey;
NSString *const UIKeyboardAnimationCurveUserInfoKey;

// Deprecated in iOS 3.2 and later.
NSString *const UIKeyboardCenterBeginUserInfoKey;
NSString *const UIKeyboardCenterEndUserInfoKey;
NSString *const UIKeyboardBoundsUserInfoKey;

```

Constants

`UIKeyboardFrameBeginUserInfoKey`

The key for an `NSValue` object containing a `CGRect` that identifies the start frame of the keyboard in screen coordinates. These coordinates do not take into account any rotation factors applied to the window's contents as a result of interface orientation changes. Thus, you may need to convert the rectangle to window coordinates (using the `convertRect:fromWindow:` (page 9) method) or to view coordinates (using the `convertRect:fromView:` method) before using it.

Available in iOS 3.2 and later.

Declared in `UIWindow.h`.

`UIKeyboardFrameEndUserInfoKey`

The key for an `NSValue` object containing a `CGRect` that identifies the end frame of the keyboard in screen coordinates. These coordinates do not take into account any rotation factors applied to the window's contents as a result of interface orientation changes. Thus, you may need to convert the rectangle to window coordinates (using the `convertRect:fromWindow:` (page 9) method) or to view coordinates (using the `convertRect:fromView:` method) before using it.

Available in iOS 3.2 and later.

Declared in `UIWindow.h`.

`UIKeyboardAnimationCurveUserInfoKey`

The key for an `NSValue` object containing a `UIViewAnimationCurve` constant that defines how the keyboard will be animated onto or off the screen.

Available in iOS 3.0 and later.

Declared in `UIWindow.h`.

`UIKeyboardAnimationDurationUserInfoKey`

The key for an `NSValue` object containing a `double` that identifies the duration of the animation in seconds.

Available in iOS 3.0 and later.

Declared in `UIWindow.h`.

`UIKeyboardCenterBeginUserInfoKey`

The key for an `NSValue` object containing a `CGPoint` that is the center of the keyboard in window coordinates before animation. These coordinates actually take into account any rotation factors applied to the window's contents as a result of interface orientation changes. Thus, the center point of the keyboard is different in portrait versus landscape orientations.

Use the `UIKeyboardFrameBeginUserInfoKey` key instead.

Available in iOS 2.0 and later.

Deprecated in iOS 3.2.

Declared in `UIWindow.h`.

UIKeyboardCenterEndUserInfoKey

The key for an `NSValue` object containing a `CGPoint` that is the center of the keyboard in window coordinates after animation. These coordinates take into account any rotation factors applied to the window's contents as a result of interface orientation changes. Thus, the center point of the keyboard is different in portrait versus landscape orientations.

Use the `UIKeyboardFrameEndUserInfoKey` key instead.

Available in iOS 2.0 and later.

Deprecated in iOS 3.2.

Declared in `UIWindow.h`.

UIKeyboardBoundsUserInfoKey

The key for an `NSValue` object containing a `CGRect` that identifies the bounds rectangle of the keyboard in window coordinates. This value is sufficient for obtaining the size of the keyboard. If you want to get the origin of the keyboard on the screen (before or after animation) use the values obtained from the user info dictionary through the `UIKeyboardCenterBeginUserInfoKey` (page 13) or `UIKeyboardCenterEndUserInfoKey` (page 14) constants.

Use the `UIKeyboardFrameBeginUserInfoKey` or `UIKeyboardFrameEndUserInfoKey` key instead.

Available in iOS 2.0 and later.

Deprecated in iOS 3.2.

Declared in `UIWindow.h`.

Notifications

UIWindowDidBecomeVisibleNotification

Posted when an `UIWindow` object becomes visible.

The notification object is the window object that has become visible. This notification does not contain a *userInfo* dictionary.

Availability

Available in iOS 2.0 and later.

Declared In

`UIWindow.h`

UIWindowDidBecomeHiddenNotification

Posted when an `UIWindow` object becomes hidden.

The notification object is the window object that has become hidden. This notification does not contain a *userInfo* dictionary.

Availability

Available in iOS 2.0 and later.

Declared In

`UIWindow.h`

UIWindowDidBecomeKeyNotification

Posted whenever a `UIWindow` object becomes the key window.

The notification object is the window object that has become key. This notification does not contain a *userInfo* dictionary.

Availability

Available in iOS 2.0 and later.

Declared In

`UIWindow.h`

UIWindowDidResignKeyNotification

Posted whenever a `UIWindow` object resigns its status as main window.

The notification object is the window object that has resigned its main window status. This notification does not contain a *userInfo* dictionary.

Availability

Available in iOS 2.0 and later.

Declared In

`UIWindow.h`

UIKeyboardWillShowNotification

Posted before a `UIWindow` object is displayed.

The notification object is `nil`. The *userInfo* dictionary contains information about the keyboard. Use the keys described in [“Keyboard Notification User Info Keys”](#) (page 12) to get the location and size of the keyboard from the *userInfo* dictionary.

Availability

Available in iOS 2.0 and later.

Declared In

`UIWindow.h`

UIKeyboardDidShowNotification

Posted after a `UIWindow` object is displayed.

The notification object is `nil`. The *userInfo* dictionary contains information about the keyboard. Use the keys described in [“Keyboard Notification User Info Keys”](#) (page 12) to get the location and size of the keyboard from the *userInfo* dictionary.

Availability

Available in iOS 2.0 and later.

Declared In

`UIWindow.h`

UIKeyboardWillHideNotification

Posted before a `UIWindow` object is hidden.

The notification object is `nil`. The `userInfo` dictionary contains information about the keyboard. Use the keys described in [“Keyboard Notification User Info Keys”](#) (page 12) to get the location and size of the keyboard from the `userInfo` dictionary.

Availability

Available in iOS 2.0 and later.

Declared In

`UIWindow.h`

UIKeyboardDidHideNotification

Posted after a `UIWindow` object is hidden.

The notification object is `nil`. The `userInfo` dictionary contains information about the keyboard. Use the keys described in [“Keyboard Notification User Info Keys”](#) (page 12) to get the location and size of the keyboard from the `userInfo` dictionary.

Availability

Available in iOS 2.0 and later.

Declared In

`UIWindow.h`

Document Revision History

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

Date	Notes
2010-05-06	Added symbols introduced in iOS 4.0.
2010-03-18	Added the screen property, which allows windows to be moved from screen to screen.
2009-03-11	Updated for iOS 3.0.
2008-09-09	Revised description of <code>UIKeyboardBoundsUserInfoKey</code> .
2008-05-01	New document that describes the class used to manipulate windows and distribute window events to the appropriate view.

REVISION HISTORY

Document Revision History