
UIAlertViewDelegate Protocol Reference

User Experience: Windows & Views



2008-05-16



Apple Inc.
© 2008 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, and iPhone 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

UIAlertViewDelegate Protocol Reference 5

Overview 5

Tasks 5

 Responding to Actions 5

 Customizing Behavior 6

 Canceling 6

Instance Methods 6

 alertView:clickedButtonAtIndex: 6

 alertView:didDismissWithButtonIndex: 6

 alertView:willDismissWithButtonIndex: 7

 alertViewCancel: 7

 didPresentAlertView: 8

 willPresentAlertView: 8

Document Revision History 9

UIAlertViewDelegate Protocol Reference

Conforms to	NSObject
Framework	/System/Library/Framework/UIKit.framework
Availability	Available in iOS 2.0 and later.
Declared in	UIAlertView.h
Related sample code	GKRocket GKTank

Overview

The `UIAlertViewDelegate` protocol defines the methods a delegate of a `UIAlertView` object should implement. The delegate implements the button actions and any other custom behavior. Some of the methods defined in this protocol are optional.

If you add your own buttons or customize the behavior of an alert view, implement a delegate conforming to this protocol to handle the corresponding delegate messages. Use the `delegate` property of an alert view to specify one of your application objects as the delegate.

If you add your own buttons to an alert view, the delegate must implement the `alertView:clickedButtonAtIndex:` (page 6) message to respond when those buttons are clicked; otherwise, your custom buttons do nothing. The alert view is automatically dismissed after the `alertView:clickedButtonAtIndex:` (page 6) delegate method is invoked.

Optionally, you can implement the `alertViewCancel:` (page 7) method to take the appropriate action when the system cancels your alert view. If the delegate does not implement this method, the default behavior is to simulate the user clicking the cancel button and closing the view.

You can also optionally augment the behavior of presenting and dismissing alert views using the methods in “Customizing Behavior” (page 6).

Tasks

Responding to Actions

- `alertView:clickedButtonAtIndex:` (page 6)
Sent to the delegate when the user clicks a button on an alert view.

Customizing Behavior

- [willPresentAlertView:](#) (page 8)
Sent to the delegate before a model view is presented to the user.
- [didPresentAlertView:](#) (page 8)
Sent to the delegate after an alert view is presented to the user.
- [alertView:willDismissWithButtonIndex:](#) (page 7)
Sent to the delegate before an alert view is dismissed.
- [alertView:didDismissWithButtonIndex:](#) (page 6)
Sent to the delegate after an alert view is dismissed from the screen.

Canceling

- [alertViewCancel:](#) (page 7)
Sent to the delegate before an alert view is canceled.

Instance Methods

alertView:clickedButtonAtIndex:

Sent to the delegate when the user clicks a button on an alert view.

```
- (void)alertView:(UIAlertView *)alertView
  clickedButtonAtIndex:(NSInteger)buttonIndex
```

Parameters

alertView

The alert view containing the button.

buttonIndex

The position of the clicked button. The button indices start at 0.

Discussion

The receiver is automatically dismissed after this method is invoked.

Availability

Available in iOS 2.0 and later.

Declared In

UIAlertView.h

alertView:didDismissWithButtonIndex:

Sent to the delegate after an alert view is dismissed from the screen.

```
- (void)alertView:(UIAlertView *)alertView
  didDismissWithButtonIndex:(NSInteger)buttonIndex
```

Parameters*alertView*

The alert view that was dismissed.

buttonIndex

The index of the button that was clicked. The button indices start at 0. If this is the cancel button index, the alert view is canceling. If -1, the cancel button index is not set.

Discussion

This method is invoked after the animation ends and the view is hidden.

Availability

Available in iOS 2.0 and later.

See Also- [alertView:willDismissWithButtonIndex:](#) (page 7)**Declared In**

UIAlertView.h

alertView:willDismissWithButtonIndex:

Sent to the delegate before an alert view is dismissed.

```
- (void)alertView:(UIAlertView *)alertView
  willDismissWithButtonIndex:(NSInteger)buttonIndex
```

Parameters*alertView*

The alert view that is about to be dismissed.

buttonIndex

The index of the button that was clicked. The button indices start at 0. If this is the cancel button index, the alert view is canceling. If -1, the cancel button index is not set.

Discussion

This method is invoked before the animation begins and the view is hidden.

Availability

Available in iOS 2.0 and later.

See Also- [alertView:didDismissWithButtonIndex:](#) (page 6)**Declared In**

UIAlertView.h

alertViewCancel:

Sent to the delegate before an alert view is canceled.

```
- (void)alertViewCancel:(UIAlertView *)alertView
```

Parameters*alertView*

The alert view that will be canceled.

Discussion

If the alert view's delegate does not implement this method, clicking the cancel button is simulated and the alert view is dismissed. Implement this method if you need to perform some actions before an alert view is canceled. An alert view can be canceled at any time by the system—for example, when the user taps the Home button. The [alertView:willDismissWithButtonIndex:](#) (page 7) and [alertView:didDismissWithButtonIndex:](#) (page 6) methods are invoked after this method.

Availability

Available in iOS 2.0 and later.

Declared In

UIAlertView.h

didPresentAlertView:

Sent to the delegate after an alert view is presented to the user.

```
- (void)didPresentAlertView:(UIAlertView *)alertView
```

Parameters*alertView*

The alert view that was displayed.

Availability

Available in iOS 2.0 and later.

See Also

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

Declared In

UIAlertView.h

willPresentAlertView:

Sent to the delegate before a model view is presented to the user.

```
- (void)willPresentAlertView:(UIAlertView *)alertView
```

Parameters*alertView*

The alert view that is about to be displayed.

Availability

Available in iOS 2.0 and later.

See Also

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

Declared In

UIAlertView.h

Document Revision History

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

Date	Notes
2008-05-16	New document that describes the UIAlertViewDelegate protocol.

REVISION HISTORY

Document Revision History